BOLT BERANEK AND NEWMAN INC

CONSULTING DEVELOPMENT • RESEARCH

BBN Report No. 3263 March 1976

COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY

Part I. Packet Radio
Part II. Speech Compression and Evaluation

Quarterly Progress Report No. 5

1 December 1975 to 29 February 1976

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935 Contract No. MDA903-75-C-0180.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: BBN Report No. 3263	2. GOVT ACCESSION NO.: AD-A155058	3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY		5. TYPE OF REPORT & PERIOD COVERED: 1 Dec. 75 - 29 Feb. 76
		6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): J.D.Burchfiel, M.D.Beeler, R.S.Nickerson, J. Makhoul, A.W.F.Huggins, R.Viswanathan		8. CONTRACT OR GRANT NUMBER(s): MDA903-75-C-0180
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: Bolt Beranek and Newman Inc. 50 Moulton St., Cambridge, Mass. 02138		12. REPORT DATE: December 1975
		13. NUMBER OF PAGES: 130
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

packet radio, computer communications, PDP-11 TCP, station gateway; ELF, BCPL, cross-radio debugging, speech compression, vocoder, linear prediction, covariance lattice, intelligibility, speech-quality evaluation, packet-loss.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This document describes progress on (1) the development of a packet radio network, and (2) speech compression and evaluation.
Activities reported under (1) include work on PDP-11 TCP development, station gateway and ELF development, and digital unit checkout; under (2) implementation of covariance lattice method; specification of ARPA-LPC System II; investigation of phoneme-specific intelligibility test; study of effects on intelligibility of lost packets.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Additional keywords: Computer communications; vocoders; linear prediction; speech quality evaluation.

BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

BBN Report No. 3263 March 1976

COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY

Part I. Packet Radio

Quarterly Progress Report No. 5
1 December 1975 to 29 February 1976

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935 Contract No. MDA903-75-C-0180.

TABLE OF CONTENTS

I. INTRODUCTION

As this quarter brings the Packet Radio Project into a new year, it also brings the development of new potentials in the station software being designed and implemented at BBN. Major progress in defining protocols to be used in the Packet Radio network provides the framework for actual communication among Packet Radio devices. Additionally, software implementation of these protocols has reached pregnant levels of function. As detailed in the section on the TCP and the gateway, considerable functional operation of those station modules has been demonstrated during this quarter. The nature of progress this quarter can roughly be described as finally having large enough and functional enough modules that we can now begin to assemble them into software that performs like a station.

At the same time, both continuation of basic support and forward looking anticipation of design issues of the future have been pursued. In the former category, maintenance of the BCPL library which supports the higher level language in which station functions are implemented has received a portion of our efforts this quarter. Also, enhancement of ELF, the operating system which provides the programming environment for the station software, has continued. In particular, timing primitives were installed to facilitate measurement of software performance. This represents a pleasant new direction in ELF support at BBN. Previously, most ELF development and support effort was required simply to obtain a

functional operating system. Now, the enhancement of ELF serves as an occasional means for bettering our software's performance and our ability to improve that performance.

In addition, this quarter includes the initiation of serious, full-time effort on the control process. This vital portion of station software has received only passing acknowledgement and vague description until now. A new member of BBN's Packet Radio group has now assimilated the history and context of the project and has become an active and important member of the group. Resolution of protocol issues has allowed substantial progress in design of the control functions to be implemented in the prototype station, as described in the section on the control process.

II. MEETINGS

On December 5 a major meeting was held at BBN for the main purpose of discussing protocol issues. The Station to Packet radio network Protocol (SPP) had been under discussion for several months. Various documents, ranging in formality from PRTNs through network messages to informal telephone discussions had provided a rich groundwork of needs and design concepts. At this meeting the various needs were compared; the means for meeting each need were compared in cost and effect on other needs and capabilities. Points of difference arising from the differing design viewpoints of the different contractors were aired. As a result of this meeting, agreement was reached on many of the issues. This is detailed in the section on the control process, since resolution of this aspect of Packet Radio network operation permitted subsequent progress on the control process.

The December 5 meeting also addressed station design, documentation, future measurement needs, and project scheduling. During this quarter several telephone conversations with Collins Radio personnel enhanced the utility of the resolutions of that meeting. Since BBN and Collins are the first implementors of the SPP protocol, this coordination permitted mutual aid and design review. We were also involved in telephone discussions with UCLA; in this case the issues were the needs for various measurements, both in general and specifically those which the control process may require for intelligent supervision of the network.

III. PUBLICATIONS

Three Packet Radio Temporary Notes were published and distributed this quarter:

PRTN 159 - "A Proposal for Incremental Routing"

PRTN 162 - "Routing in the Initial Packet Radio Network"

PRTN 165 - "Will the Real SPP Please Stand Up?"

The first of these, PRTN 159, is an outgrowth of the rich protocol development at the December 5 meeting. In large measure, PRTN 159 simply documents and solidifies ideas presented by BBN at that meeting.

As discussed in the section on the control process, reaction to and review of PRTN 159 provided an insight into SPP history and evolution. PRTN 162 was issued in an attempt to reach a new vantage point from which SPP design could be examined more globally. From this point, several alternatives became distinct; after presenting these, PRTN 162 concludes with specific recommendations about which alternatives create and preserve the maximum flexibility for the research nature of the prototype Packet Radio network. Because we feel an informed acceptance of some design strategy is essential, even if it is not composed of the alternatives we recommend, we have taken several steps to put mild pressure on our fellow contractors to review and react to this PRTN.

PRTN 165 was issued in the hope that the December 5 meeting had resolved SPP protocol issues as fully as the other members of the Packet Radio Working Group wished; that publishing the actual

specification was the only remaining task. The response to PRTN 165 proved this hope to be naive. We found that a number of design issues were misinterpreted or inappropriately applied to the network under development. We found that extensive cooperative negotiations, with SRI in particular, were necessary and, upon completion, provided fruitful basic material for another round of SPP design. While not issued as a formal publication, the text flow between the east and west coasts on this issue was considerable, and stands as a further contribution to the Packet Radio literature.

IV. STATION GATEWAY

At the beginning of this quarter, the gateway had been coded and the sections dealing with the ARPANET had been debugged. However, the sections dealing with the PR net could not be debugged until the connection process was written.

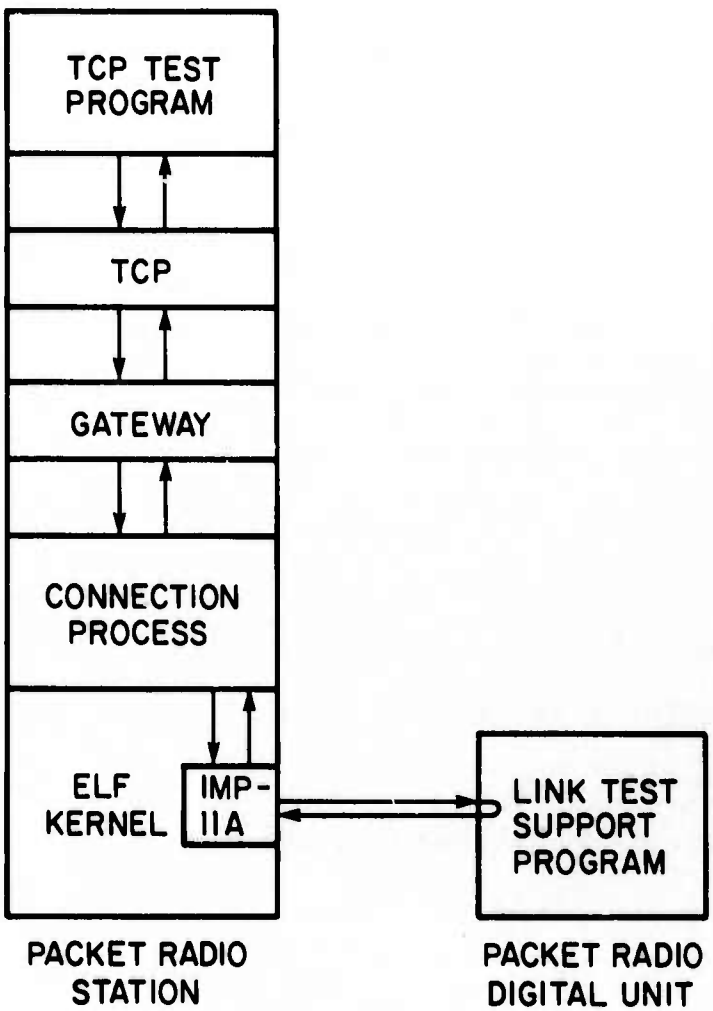
During the quarter, coding and debugging of the connection process, which implements SPP in the station, was carried on concurrently with SPP protocol discussions. The SPP protocol design was issued as PRTN #165 and after discussions with SRI and Collins in Dallas, this protocol was finalized as the protocol for use in the initial LADs.

As the connection process was altered to incorporate changes in SPP, sections of the gateway were also rewritten to conform to the current connection process implementation. After some initial debugging of the connection process, we ran the TCP, gateway and connection processes in order to debug the sections of the gateway dealing with the PRN. By the end of the quarter, we were able to demonstrate use of the gateway and connection process for PRN to PRN communications.

At this time, the interface between the connection process and the various "applications" processes -- debug, measurement, control and the gateway -- was defined. Testing of the gateway and connection processes helped to clarify this interface, and the specification is now detailed enough to allow initial

implementations of the remaining applications processes.

The configuration used for debugging the connection and gateway processes at this stage was as illustrated below. The link test support program was run in the PRDU. The connection process, gateway, TCP and TCP test program were run in the station. The TCP test program opens a connection to the PR station via a call to the TCP. Packets addressed to the station are generated by the test program and passed to the TCP which passes the packets to the gateway. On receipt of a packet for the station, the gateway calls the connection process to open a connection to the station and begins sending packets over this connection. The connection process sends the packets out through the IMP-11A interface to the PRDU where the link test support program loops the packets back to the station. On receiving a packet from the PRDU, the connection process notes from the PR header destination field that it is for the station gateway process and sends it to the gateway. The gateway notes from the internet destination fields that the packet is for the "local" Host and sends it to the TCP. The TCP returns the packets to the test program. Upon completion of all data transfers, the gateway notes that the connection is no longer in use and signals the connection process which closes the connection by sending a FIN packet. When this FIN packet is looped back to the connection process by the PRDU, the connection is closed.


TCP TEST PROGRAM
TCP
GATEWAY
CONNECTION PROCESS
ELF KERNEL
IMP-IIA
LINK TEST SUPPORT PROGRAM
PACKET RADIO STATION
PACKET RADIO DIGITAL UNIT

V. CONTROL PROCESS

The control process in the station is responsible for labeling (determining how packets are to be routed through) the network. This quarter we continued our study of the protocols governing the processing of packets by PRs (Packet Radio units) as they relate to labeling; began design and implementation of the initial version of the control process; and designed manual data entry facilities to permit exercise of other station functions in the absence of automatic labeling.

A. Protocols

The following were among the issues relevant to labeling that were resolved or clarified as a result of our December 4 meeting with Collins:

1) Terminal PRs will not forward normal traffic; thus the station must not assign routes passing through them. They will, however, relay ROps they hear to the station, so the station will have complete connectivity information available.

2) The label to be assigned to a PR will be contained in the text of the label packet, not extracted from the header. Thus the PR will not get the wrong route if the label packet is rerouted and its route overwritten.

3) A packet will be defined to unlabel a PR. This will be useful to the station for eliminating inconsistencies by reinitializing the offending PR.

4) The text of ROPs will tell whether the PR is labeled and, if so, what its labeling is.

5) PRs will never spontaneously unlabel themselves. They only become unlabeled due to manual reinitialization or receipt of an unlabel command from the station.

6) A special protocol for handling ROPs allows them to be forwarded by all PRs that hear them, not just those at a particular hierarchy level. Thus the station can assess all connectivity from a PR in a fraction of the time previously required.

7) A probe packet will be defined which the station can use to test routes. The response to the probe will tell the station what route the packet actually followed.

8) All hierarchy levels may be used (formerly one was reserved). This is a result of a new active hop acknowledgement strategy and of the use of a new header field rather than a delimiting route label to indicate the number of hops in a packet's route.

9) ROPs will contain a few performance measures made by the PR - in particular the number of inbound packets queued, alternate-routed, and dropped. The intent of this is to alert the station to problems with the first hop of a PR's route. However, since the inbound packets may not all be routed along this hop the value of these measures is questionable.

We have devoted a lot of time to the issue of what a PR knows about routing, how it knows it, and how it uses its knowledge.

At the December meeting, a change proposed by Collins was agreed to wherein PRs would not make assumptions about fixed sizes and locations of labels in a route. Instead, the field assignments would be centrally determined at the station, which would inform PRs of the location of only their own field. PRs would assume that fields appeared in order, so they could replace the remaining route of an inbound packet if desired. As before, the station would give PRs a complete route to the station.

We proposed a further change such that the station would tell PRs only a single inbound hop, not a complete route, and also the location of the inbound route field. PRs would always insert the

next hop on inbound packets. This scheme would make the measures described in (9) above refer to a single hop and would minimize the need for relabeling. This proposal was documented in PRTN 159, "A Proposal for Incremental Routing."

Critical feedback on PRTN 159 made us think more deeply about the issues of PR route knowledge. We came to feel that the design process was too haphazard: changes were being made to accomplish individual goals without understanding their effect on other goals; changes which were actually independent were being lumped together as single proposals. As a result, capabilities were being thrown away unnecessarily. We addressed these issues in PRTN 162, "Routing in the Initial Packet Radio Network." This PRTN attempted to separate the independent decisions which were made in the above proposals and show how each decision affected the capabilities of the PR and station. It ended by proposing a scheme that would retain enough flexibility for various behaviors to be tried. In particular, we recommended that the station should be able to tell a PR any amount of its route, ranging from a single hop to the whole thing, with the remainder being filled in as necessary en route; that the station should tell the PR the location of its inbound route field so the PR could make decisions based on the hop an inbound packet was taking; and that the station should tell the PR the location of the set of inbound route fields so the PR could modify the route without making assumptions about field order. This would allow features of both the Collins and BBN schemes above to be included. The recommendations of PRTN 162 are still under

consideration.

B. Control Process

Although some protocol issues still remain to be decided, enough was determined during this quarter to permit detailed design of the control process to begin. The initial version will use only those facilities that are completely understood, making simple decisions based on easily obtainable information and taking simple actions. This initial system will be described in a PRTN to be issued soon. Implementation has already begun.

C. Manual Data Entry

PRs can be given labels by direct operator input at their console terminals. We have designed and will shortly implement routines for manually informing the station of the IDs of devices in the network, the (manually-entered) labeling of PRs, and the correspondence between non-PR devices (e.g. terminals) and their attached PRs. This will enable the station to forward packets in a test network before a control process that performs automatic labeling is available.

VI. PDP-11 TCP DEVELOPMENT

The adaptation of the TENEX TCP for operation on a PDP-11 under ELF was completed during this quarter. Its proper operation was demonstrated by logging into TENEX through a user TELNET running in a PDP-11 under the ELF operating system through the PDP-11 TCP and TENEX TCP and TELNET server. A message announcing this accomplishment was sent using Mailsys to a number of interested parties. The PDP-11 TCP has also been used to transmit test data to itself using a test program which opens both ends of the connection and sends and receives a number of "letters" of data.

Preliminary measurements of the operating speed of the PDP-11 TCP indicate that it can simultaneously send and receive 5 packets per second. This figure was obtained using very short packets and measuring the amount of real time taken to transmit a given number of packets. The amount of idle time was verified to be virtually zero. The operating speed does not drop appreciably if longer packets are used indicating that the limiting factor is not due to the transfer of data from buffer to buffer.

The initial measurements were not sufficiently detailed to indicate the reason for the slow performance, so steps were taken to provide more elaborate timing measurement facilities. This required changes to both the TCP and the ELF operating system. The former to identify the CPU time required to perform various tasks within the TCP, and the latter to provide the facilities to obtain the CPU time consumed.

A new ELF primitive was added to provide the total CPU time consumed by a particular process since its creation. By taking the difference between the result of executing this primitive (CPUTM) before and after the execution of a particular task, the CPU time consumed during the execution of that task was obtained. In the process of debugging the new primitive, it was discovered that the ELF time-of-day clock did not increase monotonically. Instead, it would occasionally produce a value which was less than it should have been by a certain amount. The next reading would usually be correct. The malfunction was traced to a bug in the manner in which the hardware clock was being read. If the clock counter overflowed without being reset prior to being read, then the apparent elapsed time since the last clock reset would be small by the amount of the clock's setting. There would be no long term error, however, since the pending interrupt would take as soon as the interrupts were re-enabled and the cumulative time would be updated properly. The fix involved detecting that the overflow had occurred and adjusting the value obtained accordingly.

The debugging of the new timing facilities was completed as the quarter ended, so no definitive results were obtained, but preliminary indications are that the time consumed is distributed fairly uniformly over the various tasks. Thus the prospects are not high for obtaining a dramatic improvement. Further results will be reported next quarter.

VII. CROSS-RADIO DEBUGGER

Design and coding of the cross-radio debugger was begun this quarter. The cross-radio debugger will permit transmission of alter memory (AM) and display memory (DM) executable code packets to any selected accessible PR in the network, and provide informative printout as a function of the response to these packets. The response to a DM packet will contain the data in the specified memory locations; this will be printed on the station operator's terminal. In the event that no end to end acknowledgement is received, the cross-radio debugger will so inform the operator. In this and other respects of basic design, the cross-radio debugger is patterned after the debugging package which Collins Radio has implemented for sending AM and DM commands from a PR local console to either the PR or a remote PR.

The coding of the cross-network debugger will be completed in the next quarter, as will be its testing and inclusion in the growing collection of station software. The solidification of the interface between the connection process and a user process (the cross-network debugger in this case) late this quarter will facilitate the completion of this task.

VIII. SUPPORT SOFTWARE

A. PDP-11 BCPL Library

The library for support of BCPL programs running under ELF was partially rewritten and expanded. The rewrite was to improve the efficiency of terminal IO and to permit better interlocking of output from various processes using the same device. The expansion resulted from providing routines that call ELF primitives directly rather than using the ELFCAL function.

The number printing routines were modified to permit better control of format. This involved the addition of width and format arguments to the WriteOct, WriteN, and WriteNumber functions.

B. Other ELF Changes

In addition to the ELF changes described above, changes were also made to improve the action taken when a program running ELF executed an illegal instruction or otherwise illegally trapped. The principle problem was that the registers reported after the trap occurred were those of the kernel routine that fielded the trap rather than those of the user program executing the instruction which trapped. A secondary problem was that the program could not be restarted in any way.

This was remedied by making the routine fielding the trap take the same action as that taken when an EMT is executed. This, among other things, saves the contents of the user program's registers in

the so-called AC block. In this way, they are accessible to the cross-net debugger just as if the program had been suspended in the midst of executing an ELF primitive.

This change has subsequently facilitated the diagnosis and correction of a number of obscure bugs in the TCP and other programs.

IX. PACKET RADIO DIGITAL UNIT

During this quarter further debugging of the Packet Radio Digital Unit (PRDU) hardware problem, noticed previously, was performed. The circumstances and nature of the problem were catalogued extensively. Briefly, the problem involves the PRDU halting. Once halted, there is very little which can be determined about the state of the PRDU, which hampered debugging efforts. The halting occurs only when particular software in the PDP-11 is transmitting packets to particular software in the PRDU. The clock rate on the receive DMA in the PRDU must be within a certain critical range. At settings of delay less than the critical range, a second problem was occasionally noted. This second problem involves the PRDU hanging (no further input accepted) on the second initiation of traffic to it from the PDP-11. The final recourse was to take a complete memory dump of the affected CAP and IO routine software after the PRDU had halted, and forward this to Collins Radio for diagnosis. At about the same time that Collins personnel decided they could obtain no clues from the memory dump, the hardware was moved to a new building at BBN. After the move, the halting problem did not seem to be present, although the hangup problem still occurred occasionally. The decision was made to postpone further work on the problem by adjusting the clock delay to a large time interval, at which neither halting nor hangup occur. With this resolution, testing and provisional acceptance of the second PRDU is complete.

X. IMP-11A INTERFACE

A timing bug was found in the DEC IMP11A interface hardware which was manifested when the IMP11A was connected to the Pluribus IMP with a cable of the appropriate length and loss characteristics, and when the interface was operated in a particular manner. The problem was traced to the interface occasionally generating a short pulse (0 to 60 nsec) on the ready for next bit line going to the IMP whenever the word count was exhausted without receiving a last bit signal from the IMP. This usually occurred when running the network bootstrap program but not during normal operation. It furthermore required the slightly higher speed logic of the Pluribus IMP and a cable that would transport the pulse to the IMP at the proper time. The pulse originated in a hazard between two signals making a transition caused by the the same source. The "or" of the two signals was used to prevent the ready for next bit signal coming on. The cure was to generate a signal equivalent to the one required but without any holes in it.

This modification has been given to DEC for inclusion in subsequent IMP11A interfaces and for distribution to other users of the interface.

BBN Report No. 3263 March 1976

COMMAND AND CONTROL RELATED COMPUTER TECHNOLOGY

Part II. Speech Compression and Evaluation

Quarterly Progress Report No. 5
1 December 1975 to 29 February 1976

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 2935 Contract No. MDA903-75-C-0180.

TABLE OF CONTENTS

I. INTRODUCTION

In the last quarter, we developed a new formulation for linear prediction, which we call the covariance lattice method. The method is one of a class of lattice methods which guarantee the stability of the all-pole linear prediction filter, with or without windowing of the signal, with finite wordlength computations and with the number of computations being comparable to the traditional autocorrelation and covariance methods. We incorporated the covariance lattice method into our floating-point simulation of the LPC speech compression system. This also involved "tuning" of such quantities as analysis interval and criterion for determining optimal LPC order, to obtain approximately the same speech quality as that from our earlier 1500 bps LPC system (which uses the autocorrelation method) at about the same total computational time. In fixed-point implementations, however, the guaranteed filter stability provided by the covariance lattice method might lead to an improvement in speech quality relative to that from the autocorrelation LPC system.

We presented a summary of major results of our speech compression project in the last 3 years at the December ARPA Review Meeting. This summary was also issued as NSC Note 77 and is reproduced in this report as Appendix A.

Also in the last quarter, we provided specifications for ARPA-LPC speech compression system II, an update of the present system I. The system II as specified by us will be implemented at

the different ARPA-sponsored sites.

In our work on quality evaluation this quarter, we have run a phoneme-specific intelligibility test on a subset of five of the fourteen LPC-vocoder systems we studied earlier. The analysis of the results of this experiment is nearly complete. We have also analyzed the effects of lost or delayed packets on speech intelligibility, and suggested a modified way of packetizing speech so as to minimize the intelligibility decrement. The suggestion, together with the arguments leading up to it, was issued as NSC Note #78, and is reproduced in this report as Appendix D.

II. COVARIANCE LATTICE METHOD FOR LINEAR PREDICTION

The covariance lattice method is a hybrid between the covariance method and traditional lattice methods. The new method has all the advantages of a regular lattice, plus the added advantage of a computational efficiency comparable to the non-lattice methods.

As mentioned in the introduction, the covariance lattice method is one of a class of lattice methods with many desirable properties. The formulation of these lattice methods and their efficient computational procedure are described in NSC Note 75, a copy of which is attached with this report as Appendix B.

A program with spectral and waveform display capabilities was written for use from our IMLAC PDS-1 display terminal to experimentally study the covariance lattice method. Using this program, we verified experimentally the results analytically established in Appendix B. As expected, for cases where the covariance method produced an unstable linear prediction filter, the covariance lattice method produced a stable filter. In addition, the power spectrum of the stable filter was found to be a reasonably good fit to the envelope of the short-term signal spectrum. A comparative study indicated that the covariance lattice method resulted in estimates of pole bandwidths generally larger than those obtained from the covariance method and generally smaller than those given by the autocorrelation method.

Another study that we conducted using the interactive display program was concerned with the length of the analysis interval for the covariance lattice method. Longer intervals mean more computations required in solving for the predictor parameters. With analysis intervals shorter than a pitch period, the accuracy of the power spectrum of the resulting linear predictor (relative to the envelope of the short-term speech spectrum) was found to critically depend on the location of the analysis interval relative to the pitch pulses. Notice that an analysis scheme that requires positioning of the analysis interval with respect to the location of pitch pulses is basically a pitch-synchronous scheme. Since we have not yet resolved all the issues relating to such frame positioning and since we wish to keep the analysis simple for vocoder application, we chose to employ a sufficiently long analysis interval.

Our next step was to incorporate the covariance lattice method into our floating-point simulation of the LPC vocoder. The introduction of the new analysis scheme necessitated the "tuning" or adjustment of a number of other parameters. They were: 1) length of the analysis interval, 2) criterion to determine optimal predictor order, 3) log likelihood ratio threshold used in variable frame rate transmission, and 4) bit allocation for log area ratios. The goal was to obtain approximately the same speech quality as that from our earlier 1500 bps LPC system at about the same total computational time and, of course, at the same average bit rate.

Except for the second variable, the other 3 variables mentioned above need no explanation. The information criterion that we use for selecting the predictor order is (see p. 23 of BBN Report No. 2976) the sum of the logarithm of the normalized prediction error and a linear term proportional to predictor order. For the autocorrelation method, satisfactory results were obtained when the slope of this linear term was 5/N, where N is the number of samples in the analysis window. Since the covariance lattice method does not require any windowing, the slope need be only 2/N. However, this choice of the slope yielded relatively high values for predictor order, thus increasing the bit rate. Therefore, we decided to search for a suitably large value for the slope.

The four variables given above are not independent of each other in terms of achieving the stated goal. This necessitated a large number of synthesis experiments using a broad range of speech material. Except for these four variables, all other analysis, transmission and synthesis conditions used were the same as in our earlier 1500 bps LPC system described in BBN Report No. 2976. Informal listening tests were used to judge the speech quality in these experiments. As a result of these experiments, we chose the following parameters: Analysis interval = 12.9 msec (with an initial condition of pmax = 11 samples, a total of 140 samples were used in computing the covariances defined by equation (13) of Appendix B); Slope of the linear term in the information criterion for predictor order selection = 3/N ; Log likelihood ratio threshold = 2 decibels; Variable step size quantization of log area ratios was employed with the bit (or level) allocation as given in Table 1.

Table 2 lists the average bit rates for 5 different systems. System 5 was found to produce good quality speech, approximately the same as our earlier 1500 bps system, at about the same total computational time.

In fixed-point implementations, finite wordlength computations can cause filter instabilities with the autocorrelation method. The covariance lattice method still guarantees filter stability as stated earlier. Therefore, in fixed-point implementations, the covariance lattice method might yield better quality speech than the autocorrelation method. Furthermore, as stated in Appendix B, the covariance lattice method permits the quantization of the reflection coefficients to be accomplished within the recursion for retention of accuracy in representation. Such a quantization method might also lead to an improvement in the quality of the synthesized speech.

III. SPECIFICATIONS FOR ARPA-LPC SYSTEM II

The approach we employed in arriving at the specifications was to reap maximum benefit for the least amount of effort in terms of changes to the present System I. Our overall design objective was to achieve average continuous-speech transmission rates of about 2200 bps. With the use of a silence detection algorithm, these rates may drop to about 1000 bps or less.

There are two major differences between System I and II. These are: 1) Variable frame rate transmission of LPC parameters, and 2) use of new coding/decoding tables for transmission parameters. The details of System II specifications are contained in NSC Note 82 which is included in this report as Appendix C.

IV. REAL-TIME IMPLEMENTATION

We moved the SPS-41/PDP-11 system into our new building. We found and fixed several hardware failures and installation errors. The system currently runs the back-to-back LPC program for 3 to 4 hours before failing.

We plan to develop an operating system for our SPS-41/PDP-11 facility. We will then generate necessary software for A/D and D/A spooling.

V. PHONEME-SPECIFIC INTELLIGIBILITY TESTS

A. Purpose

If two communications systems differ noticeably in intelligibility, the question of their relative quality rarely arises. As a result, quality comparisons are usually performed only on sets of systems that have equal (and usually high) intelligibility. It has often been argued that the information obtained from quality tests could better be obtained from intelligibility tests, if the latter could only be made sufficiently difficult that the scores dropped substantially below 100%. As an extreme example, consider a pair of systems that both score 98% on Intelligibility Test 'X'. Test 'X' is based on measuring the intelligibility of a two-word vocabulary, consisting of the digits 'one', and 'two'. It is obvious that there might be considerable differences in the quality of the speech passed by the two systems that test 'X' would fail to detect. On the other hand, a more difficult test, based perhaps on PB word lists, might well separate the two systems.

The question of whether quality tests and intelligibility tests are measuring the effects of the same variables is a very important one. Quality tests are much more subjective than intelligibility tests, since they require the subject to make a judgment, such as a rating or a preference, for which there is no objectively correct response. Consequently, the results of quality tests are heavily dependent on the set of systems being compared, on the test subjects

and the instructions they are given, and on a variety of other variables that are hard to control and hard to quantify. Nakatani and Dukes (1971) have had some success in showing the equivalence between quality measures and their 'Q-Measure' of intelligibility, but unfortunately their procedure is complicated and expensive to run. Furthermore, the quality data against which Nakatani and Dukes compared their Q-Measure results were much less rich in detail than the quality data available to us, as a result of the quality tests we have reported in earlier QPR's. Since the results of our tests were successful in providing diagnostic information about how the vocoders differed in quality, it was considered important to use an intelligibility test that was capable of yielding similar diagnostic detail. This permits a much more detailed comparison of the two methods than if a simple percent-correct test were used. For example, it makes possible the use of the same multi-dimensional scaling procedures for analyzing both sets of data. The results of the analyses can then be compared, to see if the results are well described by a single psychological structure. This is a procedure we have already had some success with, as described in BBN Report No. 3209, where we showed that the rank-ordering task and the rating task, produce highly similar results in quality evaluation.

B. The Phoneme-Specific Intelligibility Test

The phoneme-specific intelligibility test we adopted is a development of one described by Stevens (1962). The test has two

parts, one for consonants and one for vowels. It is a nonsense-syllable test, using closed response sets of 4-8 items. Both of these factors increase the difficulty of the test over that of the Diagnostic Rhyme Test (DRT: Voiers et al, 1973), which is the only other test available with similar diagnostic power. The DRT measures only consonants, and only in initial position, and the response set for each item is a minimal pair of English monosyllables. The Phoneme-Specific Intelligibility test covers vowels and consonants in both pre-stress and in final position. The stimulus items are nonsense syllables of the form /ə'C1VC2/, where /ə/ is an unstressed schwa like the first syllable of 'about', C1 and C2 are consonants, and V is a stressed vowel. The complete test consists of 14 separate subtests. The first ten are consonant tests, each of which uses a single closed set of consonants from which C1 and C2 are drawn. There are four versions of each consonant subtest, two of which use one pair of vowels as syllable nuclei, and two using a second pair of vowels. A typical consonant test list is shown in Figure 1. Each consonant in the closed response set appears four times in each list, once preceding and once following each of the two context vowels. In addition, there are three filler items (ringed numbers in Figure 1) added to prevent subjects from using the symmetry of the test to aid their responding. The vowel tests are similar, except that each vowel appears four times in each list, in symmetrical consonant context, and there are three different sets of consonant contexts for each vowel subtest. The complete test is summarized in Table 1, which

IBM

TEST NO. _____ NAME _____ DATE _____

CONSONANTS: b d g k p t

VOWELS: a ɪ

1. p a b
2. b ɪ t
3. g ɪ t
4. t a g
5. p ɪ b
6. k ɪ d
7. d ɪ g
8. g a d
9. d a k
10. t ɪ p
11. b ɪ k
12. b a t
13. d a p
14. k a k
15. p a k

Figure 1: A sample consonant test list. Each nonsense syllable is preceded by an unstressed vowel, and contains an initial and final consonant drawn from the consonant response set, and a vowel from the context vowel set. The ringed items are fillers.

gives the response set and context sets for each of the ten consonant subtests, and for each of the four vowel subtests.

C. Talkers and Recordings

Two talkers each recorded one of the symmetrical halves of the complete test. All lists with an 'M' in the title (See Table 3) were read by the male talker, who had a low fundamental. (He was speaker #3, DK, in the quality tests). The lists with an 'F' in the title were read by a female talker. Both had considerable experience with phonetic symbols, and with recording techniques. The lists were read in a sound-treated room, and were recorded with a boom-mounted electret microphone (Thermo Electron, Model 5336), and high-quality recording equipment. The items in a list were read at a constant vocal effort, and at a rate of one item every 5.5 seconds, cued by a flash of light from an electronic interval timer. Errors and slurred productions were removed by repeating the whole list. It took approximately three hours to record each talker.

D. Selection of Lists and Systems for Pilot Experiment

Although all the 64 lists in the complete test were recorded, the amount of material involved precludes using the complete test, except for testing real-time systems. To keep the experiment within reasonable proportions, we selected seven consonant lists from the total of 64, and five of the computer-simulated vocoder systems from

the 14 used in our earlier quality tests. Six of the selected lists were from the set spoken by the male speaker, and one was spoken by the female speaker. The reasons for choosing only consonant lists were:

1. The consonant lists are intrinsically harder than the vowel lists, partly because most of them require two responses per item.

2. The vowel tests require of the subjects a greater familiarity with phonetic symbols for writing down their responses, and we wished to avoid lengthy training sessions.

The lists we selected are underlined in Table 3. They consist of lists 1BM, 2AM, 3BM, 4BM, 7AM, and 10AM spoken by the male talker, and list 7BF spoken by the female.

In addition to the 9-bit PCM, unvocoded version of each test list, the seven lists were processed through four vocoder systems. These selected systems were systems A, D, F and G in BBN Report No. 3209, which were all fixed-rate systems, so that their bit rates did not vary with the speech material.

The vocoders include one of the best, one of the worst, and two other systems whose relative quality depended heavily on the speech materials.

E. Procedure

In our first pilot experiment, we presented the 35 processed lists (7 lists x 5 systems) in an irregular order to a group of listeners. It soon became obvious, however, that error rates were

low, and that subjects became aware that the same lists were being repeated several times. For these two reasons we redesigned the pilot experiment to correct these deficiencies.

First, by cutting and splicing the stimulus tapes, we arranged that in each of the five presentations of a list, one through each system, the list appeared in a different cyclic permutation. Secondly, subjects were run in groups of four, and although each group of subjects heard all 35 processed lists, in the same cyclic order, each group started in a different place in the cyclic order. Thus, each of the five versions of a given list was heard in the first block of seven lists by one group of subjects, in the second block of seven by a second group of subjects, and so on. This effectively counterbalanced the presentation order, and controlled for learning effects.

Thirdly, a revised response sheet was composed for each test list, as shown in Figure 2, and a secondary task was introduced, so that correct items as well as errors would yield data on the relative intelligibility of the systems. The secondary task was to write down, after each item, the number appearing on a digital counter in front of the subjects. The clock count incremented every 100 msec, and the count was reset to zero by the experimenter at the instant of presentation of each stimulus item. Thus the subjects were, in effect, recording a rather gross measure of the time they had taken to make each response.

Name _____

LIST # 1

CONSONANTS: b d g k p t

VOWELS: a (father)
I (bit)

(1.)	b d g k p t	--a--	b d g k p t	------
(2.)	b d g k p t	--I--	b d g k p t	------
(3.)	b d g k p t	--I--	b d g k p t	------
(4.)	b d g k p t	--a--	b d g k p t	------
(5.)	b d g k p t	--I--	b d g k p t	------
(6.)	b d g k p t	--I--	b d g k p t	------
(7.)	b d g k p t	--I--	b d g k p t	------
(8.)	b d g k p t	--a--	b d g k p t	------
(9.)	b d g k p t	--a--	b d g k p t	------
(10.)	b d g k p t	--I--	b d g k p t	------
(11.)	b d g k p t	--I--	b d g k p t	------
(12.)	b d g k p t	--a--	b d g k p t	------
(13.)	b d g k p t	--a--	b d g k p t	------
(14.)	b d g k p t	--a--	b d g k p t	------
(15.)	b d g k p t	--a--	b d g k p t	------

Figure 2: A sample response sheet. The subject marks one of the initial consonants (left) and one of the final consonants (right).

F. Subjects

The twenty subjects were students at a local High School that responded to an advertisement. They served in groups of four, and were paid for their services. The experiment was run in a quiet room, and the stimulus tapes were played through a high quality loud speaker. The instructions that were read to the subjects are presented in Appendix E. Several practice items were given, and care was taken to make sure the subjects understood the task. The whole experiment took about 2 hours, including several rests.

G. Results: Overall Error Rates

We present below a summary of the distribution of errors, as a function of the test list, and the vocoder system it was processed through. We also present confusion matrices, for each list and system, although we will postpone detailed discussion of these until a later report. Our analyses of the response-time data from the secondary task are not yet complete, nor have we made comparisons between the results of the present intelligibility tests and the earlier quality tests.

The most gross summary of errors is presented in Table 4, which shows the total number of errors made by the 20 subjects, categorized by the test list and by the vocoder system the list was processed through. The error totals are further broken down by whether the error occurred on an initial or a final consonant.

The total error rate across all systems and all lists was 9.14% (a total of 1463 errors out of a possible 16,000). The total error rate across all lists varied from 4.7% for the PCM unvocoded speech to 12.6% for system F (10-poles, 25 msec frame size, 0.2 dB quantization step size). The other three systems all generated error rates close to 9.5%. Pooled across all systems, the error rates on the different lists varied from 3.7% on list 10AM (initial stop clusters) to 15.7% on list 4BM (voiced and voiceless fricatives). This range of total error rates was considerably smaller than we had hoped: it appears that this test is not sufficiently difficult to separate the systems very widely. An alternative method to increase the difficulty of the tests is to record the test materials under degraded conditions. The major problem with this approach is reproduceability, since simply adding noise is not very realistic. It is also important not to lose sight of the conditions under which the vocoding system will actually be used. If the problem is to select one of a pair of vocoder systems, for use in quiet offices, the results of comparing them in 100 dB aircraft noise is not likely to be very relevant -- yet it may be necessary to degrade recording conditions this much to get a significant difference between the systems.

The overall error scores in Table 4 are not very informative. For initial consonants and for final consonants, and for both combined, System N (PCM Speech) produced the fewest errors, and System F produced the most. We have not yet completed a careful comparison of the present results with those of the earlier quality

tests, but in those tests, System G was found to have consistently worse quality than System F. Thus, at first sight it appears that the quality results may be different from the intelligibility results. It is interesting to note that, in the one list recorded with a female voice, List 7BF, System G yielded the fewest errors -- fewer even than System N, the PCM original. This result does not seem very likely -- it may be due to lack of balance between the five groups of experimental subjects.

Table 5 presents the same error data as Table 4, this time further broken down by each phoneme in the response set. Each cell represents the number of errors made by twenty subjects, to two presentations of the specified phoneme (three presentations for final m, ng, in List 7AM; and final m, r, in List 7BF). Thus cell totals are 40 (60 for the foregoing exceptions).

Inspection of Table 5 shows that a few phonemes accounted for a large number of errors. For example, in List 2AM, /k/ in initial position yielded 20-22 errors for each of the vocoder systems except N (PCM speech). Inspection of the individual subjects response sheets shows that subjects were in strong agreement on their errors: of the total of 84 errors, 68 of the initial k's were heard as p's, and 14 were heard as f's. It is possible that this high degree of agreement was due to a response bias, induced perhaps by earlier items in the list. Other examples that may have a similar explanation occurred in List 3AM for initial /g/ (55 out of 56 g's were heard as v's); in List 4BM for initial /zh/ (here the errors

may be due to subjects lack of familiarity with the discrimination required -- they are distributed over all systems, including N, the PCM speech) and for final /s/ (59 out of 87 errors heard as z); and for final /m/ in Lists 7AM and 7BF (83 out of 105, and 56 out of 61 being heard as ng, respectively). The overall error rates would be considerably lower if these errors were ignored. However, it should be noted that few of these errors occurred with system N (PCM speech) -- in other words, they only occurred when the speech was somewhat degraded by the vocoder system.

Tables 6 - 12 give an even more detailed break-down of the errors for each list in the confusion matrices. We will postpone detailed discussion of these until we have made the comparisons with the results of the quality tests. The analysis of the reaction time data will also be available by then.

VI. TABLES 1-12

Table 1. Number of quantization levels for log area ratios.

COEFF. #	1	2	3	4	5	6	7	8	9	10	11
VOICED	33	25	19	14	13	10	11	10	8	8	7
UNVOICED	40	22	14	12	10	8	13	8	8	7	6

Table 2. Average bit rates for 5 LPC systems.

SYSTEM #	Variable Frame Rate	Variable Order	Optimal Linear Interpolation	Huffman Coding	Bit Rate (bps)
1	NO	NO	NO	NO	4520
2	YES	NO	NO	NO	1920
3	YES	YES	NO	NO	1750
4	YES	YES	YES	NO	1800
5	YES	YES	YES	YES	1525

Table: 3

A. Consonant Tests

List ID	Context Vowels	Response Set	# in list
1AM,1AF/1BM,1BF	u, ɛ / ɑ, ɪ	p,t,k,b,d,g	15
2AM,2AF/2BM,2BF	i, ʊ/ ɑ, ɛ	p,t,k,f,s,sh	15
3AM,3AF/3BM,3BF	æ, ʊ/ ɑ, ɪ	b,d,g,v,z,zh	15
4AM,4AF/4BM,4BF	u, ɪ / æ, ʌ	f,s,sh,v,z,zh	15
5AM,5AF/5BM,5BF	æ, ʌ / u, ɛ	b,d,m,n,v,z	15
6AM,6AF/6BM,6BF	i, ʊ / ɑ, ɛ	ch,j,s,sh,z,zh	15
7AM,7AF/7BM,7BF	ɑ, i / ɑ,æ	l,r,w,y,m,n (*) l,r,m,n,ng (**)	15
8AM,8AF/8BM,8BF	æ, ʌ / u, ɪ	f,s,sh,θ	11
9AM,9AF/9BM,9BF	aɪ, ɛ / ɑ,ɛ	d,l,n,r,ld,nd,rd (final clusters)	15
10AM,10AF/10BM,10BF	i, ɑ / o, e	s,sw,sl,sm,sn,sp,st,str (initial clusters)	19

Vowels: i=beet, I=bit, ɛ=bet, æ=bat, a=father, ʌ=cup, o=go, e=bait, aɪ=bite, ʊ=foot, u=food.

(*) = Initial
(**) = Final

B. Vowel Tests

List ID	Context Consonants	Response Set	# in List
11AM/11AF	b d m w		19
11BM/11BF	m w p t	ɪ, ɛ, ʌ, ʊ	19
11CM/11CF	f s v z		19
12AM/12BF	b d m n		19
12BM/12BF	m n p t	i, ɪ, ɛ, æ	19
12CM/12CF	f s v z		19
13AM/13AF	b d m n		19
13BM/13BF	m n p t	u, ʊ, ʌ, ɑ	19
13CM/13CF	f s v z		19
14AM/14AF	b d m n		19
14BM/14BF	m n p t	i, æ, ɑ, u	19
14CM/14CF	f s v z		19

Table: 4

Resp set	ptk bdg	fs,sh vz,zh	ptk fs,sh	bdg vz,zh	init: fin:	lmnrwy lrmn,ng	init clust		
List:	1BM	4BM	2AM	3BM	7AM	7BF	10AM	Tot	%
Initial Errors									
System									
N	7	34	10	11	4	11	5	82	4.66
A	21	41	31	22	8	7	2	132	7.50
D	14	35	39	41	10	6	14	159	9.03
F	11	43	37	57	14	15	33	210	11.9
G	22	45	37	25	14	3	5	151	8.58
Final Errors									
System									
N	9	20	13	7	3	17		69	4.79
A	21	48	22	15	24	26		156	10.83
D	17	28	18	28	37	25		153	10.63
F	20	50	18	34	44	28		194	13.47
G	27	33	26	23	30	18		157	10.90
Initial + Final Errors									
System									
N	16	54	23	18	7	28	5	151	4.72
A	42	89	53	37	32	33	2	288	9.00
D	31	63	57	69	47	31	14	312	9.75
F	31	93	55	91	58	43	33	404	12.63
G	49	78	63	48	44	21	5	308	9.63
Total:	169	377	251	263	188	156	59	1463	
%:	7.04	15.71	10.46	10.96	7.80	6.50	3.69	9.14	

Table: 5a Error Summaries

LIST 1BM:

INITIAL						FINAL					
SYS: STIM	N	A	D	F	G	SYS:	N	A	D	F	G
B		6		1	5	B	4	9	1	3	4
D		1	1		1	D		1			3
G	2	4	3	2	6	G		2	1	1	4
P	1	1	2			P	1	1	7	6	4
T	1	4	7	5		T	2	6	6	5	6
K	3	5	1	3	10	K	2	2	2	5	6
-TOT-	7	21	14	11	22	-TOT-	9	21	17	20	27

LIST 2AM:

INITIAL						FINAL					
SYS: STIM	N	A	D	F	G	SYS:	N	A	D	F	G
P						P	2	7	3	5	7
T			4	4		T	3	2	1	2	2
K	3	21	20	22	21	K		5	7	7	9
F		1	1	2	4	F	6	2	1		2
S	2	5	8	4	8	S	1	4	1	1	1
SH	5	4	6	5	4	SH	1	2	5	3	5
-TOT-	10	31	39	37	37	-TOT-	13	22	18	18	26

LIST 3BM:

INITIAL						FINAL					
SYS: STIM	N	A	D	F	G	SYS:	N	A	D	F	G
B		13	17	26	4	B	1	5	5	15	2
D			5	18	1	D	1	2	6	9	5
G			2	1	1	G		1	3	3	6
V		1	2		2	V		3	5		3
Z	1	1	2	3	1	Z	1		1	1	1
ZH	10	7	13	9	16	ZH	4	4	8	6	6
-TOT-	11	22	41	57	25	-TOT-	7	15	28	34	23

LIST 4BM:

INITIAL						FINAL					
SYS: STIM	N	A	D	F	G	SYS:	N	A	D	F	G
F	1	8	5	4	7	F	2	4	2	10	3
S	6	4	6	10	9	S	10	22	14	25	16
SH	6	8	4	3	3	SH	2	3	3	4	3
V	4	2	6	3	4	V		2	2	2	4
Z	4	4	6	8	8	Z	1	7	1	5	
ZH	13	15	8	15	14	ZH	5	10	6	4	7
-TOT-	34	41	35	43	45	-TOT-	20	48	28	50	33

Table: 5b Error Summaries

LIST 7AM:

	INITIAL					FINAL					
SYS: STIM	N	A	D	F	G	SYS:	N	A	D	F	G
L		4	5	7	5	L				1	
R			1	3	5	R					1
W			2	1	3	M		18	29	36	22
Y	4	2		1		N	2	6	6	7	6
M			2	1	1	NG	1		2		1
N		2		1							
-TOT-	4	8	10	14	14	-TOT-	3	24	37	44	30

LIST 7BF:

	INITIAL					FINAL					
SYS: STIM	N	A	D	F	G	SYS:	N	A	D	F	G
L	2			1		L	3		3	2	1
R	2	2	1	2	1	R					1
W	2	2	1	7	1	M	9	20	11	11	10
Y	3	1		2		N	4	6	6	5	3
M	1	1	3	3	1	NG	1		5	10	3
N	1	1	1								
-TOT-	11	7	6	15	3	-TOT-	17	26	25	28	18

LIST 10A:

	INITIAL				
SYS: STIM	N	A	D	F	G
S	4	1	5	6	2
SL			5	12	3
SW					
SM	1		1	4	
SN		1	1	6	
SP				1	
ST				1	
STR			2	3	
-TOT-	5	2	14	33	5

Table 6:

CONFUSION MATRICES FOR LIST: 1BM

SYSTEM		INITIAL								FINAL							
	S:R	B	D	G	P	T	K	X		S:R	B	D	G	P	T	K	X	
N	B	40								B	36	2		2				4
	D		40							D		40						
	G			38			2		2	G			40					
	P	1			39				1	P	1			39				1
	T		1			39			1	T				1	38	1		2
	K			3			37		3	K			2			38		2
N	TOTAL ERRORS								7									9
A	B	34	3	1	2				6	B	31	5	2	2				9
	D		39	1					1	D		39	1					1
	G			36	2		2		4	G		1	38		1			2
	P	1			39				1	P	1			39				1
	T		2		1	36	1		4	T		1	2		34	3		6
	K		1		2	2	35		5	K			2			38		2
A	TOTAL ERRORS								21									21
D	B	40								B	39			1				1
	D	1	39						1	D		40						
	G		1	37			2		3	G		1	39					1
	P	2			38				2	P	4		1	33	1		1	7
	T	1	1		2	33	2	1	7	T			1		34	5		6
	K			1			39		1	K	1			1		38		2
D	TOTAL ERRORS								14									17
F	B	39						1	1	B	37			3				3
	D		40							D		40						
	G	1		38			1		2	G	1		39					1
	P				40					P	2	1	2	34		1		6
	T		3		2	35			5	T				1	35	3	1	5
	K			2	1		37		3	K	1		1	3		35		5
F	TOTAL ERRORS								11									20
G	B	35			5				5	B	36	2		1		1		4
	D		39	1					1	D		37	2				1	3
	G			34	3		2	1	6	G		3	36			1		4
	P				40					P	2			36	1	1		4
	T					40				T		1	2		34	3		6
	K				7	3	30		10	K			4	2		34		6
G	TOTAL ERRORS								22									27

Table 7:

CONFUSION MATRICES FOR LIST: 2AM

SYSTEM		INITIAL							FINAL							
	S:R	P	T	K	F	S	SH	X	S:R	P	T	K	F	S	SH	X
N	P	40							P	38		1				1 2
	T		40						T		37	1		1		1 3
	K		1	37	1			1 3	K			40				
	F				40				F	2		3	34		1	6
	S					38	2	2	S					39	1	1
	SH					5	35	5	SH					1	39	1
N	TOTAL ERRORS							10								13
A	P	40							P	33	3	4				7
	T		40						T	1	38	1				2
	K	20	1	19				21	K	2	1	35	2			5
	F		1		39			1	F			1	38		1	2
	S					35	5	5	S	1			1	36	2	4
	SH					4	36	4	SH					2	38	2
A	TOTAL ERRORS							31								22
D	P	40							P	37		1	2			3
	T	4	36					4	T	1	39					1
	K	13	1	20	6			20	K	1		33	6			7
	F	1			39			1	F				39			1
	S					32	8	8	S	1				39		1
	SH		1			5	34	6	SH					5	35	5
D	TOTAL ERRORS							39								18
F	P	40							P	35		4	1			5
	T	3	36	1				4	T	2	38					2
	K	14		18	8			22	K	1		33	6			7
	F	2			38			2	F				40			
	S					36	4	4	S					39	1	1
	SH					5	35	5	SH					3	37	3
F	TOTAL ERRORS							37								18
G	P	40							P	33	2	2	3			7
	T		40						T		38	1	1			2
	K	21		19				21	K	1		31	8			9
	F	3		1	36			4	F	1		1	38			2
	S					32	8	8	S					39	1	1
	SH					4	36	4	SH					5	35	5
G	TOTAL ERRORS							37								26

Table 8:

CONFUSION MATRICES FOR LIST: 3BM

SYSTEM		INITIAL								FINAL							
	S:R	B	D	G	V	Z	ZH	X		S:R	B	D	G	V	Z	ZH	X	
N	B	40								B	39			1				1
	D		40							D		39	1					1
	G			40						G			40					
	V				40					V				40				
	Z					39	1		1	Z				1	39			1
	ZH			4		6	30		10	ZH			2		2	36		4
N	TOTAL ERRORS								11									7
A	B	27			13				13		B	35	2	2	1				5
	D		40								D		38	2					2
	G			40							G			39			1		1
	V	1			39				1		V		2		37	1			3
	Z				1	39			1		Z					40			
	ZH			4		3	33		7		ZH			2		2	36		4
A	TOTAL ERRORS								22										15
D	B	23			17				17		B	35	1		4				5
	D		35	5					5		D		34	4	2				6
	G			38	2				2		G		1	37		1		1	3
	V	2			38				2		V		1	1	35	1		2	5
	Z			1	1	38			2		Z					39	1		1
	ZH			6		7	27		13		ZH			3	1	4	32		8
D	TOTAL ERRORS								41										28
F	B	14			25			1	26		B	25	3	1	9		1	1	15
	D		22	10	3		4	1	18		D		31	9					9
	G			39	1				1		G		2	37		1			3
	V				40						V				40				
	Z				1	37		2	3		Z					39	1		1
	ZH			4		4	31	1	9		ZH			1		4	34	1	6
F	TOTAL ERRORS								57										34
G	B	36			4				4		B	38	1		1				2
	D		39					1	1		D		35	2		1		2	5
	G	1		39					1		G		5	34	1				6
	V	1		1	38				2		V	3			37				3
	Z					39	1		1		Z					39	1		1
	ZH			6	3	4	24	3	16		ZH			3		3	34		6
G	TOTAL ERRORS								25										23

Table 9

CONFUSION MATRICES FOR LIST: 4BM

SYSTEM		INITIAL								FINAL							
	S:R	F	S	SH	V	Z	ZH	X		S:R	F	S	SH	V	Z	ZH	X	
N	F	39			1				1	F	38			2				2
	S		34	3		2	1		6	S		30	6		2	2		10
	SH		6	34					6	SH			38			2		2
	V	3			36	1			4	V				40				
	Z		2	2		36			4	Z				1	39			1
	ZH		2	6		5	27		13	ZH			3		2	35		5
N	TOTAL ERRORS								34									20
A	F	32			8				8	F	36			3			1	4
	S		36	2		1	1		4	S		18	5		16	1		22
	SH		4	32			4		8	SH			37			3		3
	V	2			38				2	V	1		1	38				2
	Z		2	1		36		1	4	Z		2		3	33	2		7
	ZH		1	6	1	7	25		15	ZH			7	1	2	30		10
A	TOTAL ERRORS								41									48
D	F	35			4		1		5	F	38			2				2
	S		34	3	1	1		1	6	S		26	2		11	1		14
	SH		3	36			1		4	SH		1	37			2		3
	V	4	1		34	1			6	V	1		1	38				2
	Z		5	1		34			6	Z		1			39			1
	ZH			6		2	32		8	ZH			3		2	34	1	6
D	TOTAL ERRORS								35									28
F	F	36			4				4	F	30			9			1	10
	S		30	3		5	2		10	S		15	5		19	1		25
	SH		3	37					3	SH			36			4		4
	V	2			37	1			3	V	1			38		1		2
	Z		7		1	32			8	Z		1		1	35	2	1	5
	ZH			6	2	7	25		15	ZH				1	3	36		4
F	TOTAL ERRORS								43									50
G	F	33	1	1	5				7	F	37			3				3
	S	1	31	3	1	4			9	S		24	5		11			16
	SH		3	37					3	SH		1	37			2		3
	V	3			36	1			4	V	1		2	36	1			4
	Z	1	6	1		32			8	Z					40			
	ZH	1		6		7	26		14	ZH		1	1		4	33	1	7
G	TOTAL ERRORS								45									33

Table 10:

CONFUSION MATRICES FOR LIST: 7AM

SYSTEM		INITIAL								FINAL							
	S:R	L	R	W	Y	M	N	X		S:R	L	R	M	N	NG	X	
N	L	40								L	40						
	R		40							R		40					
	W			40						M			60				
	Y		1	2	36			1	4	N			2	38			2
	M					40				NG					59	1	1
	N						40										
N	TOTAL ERRORS								4								3
A	L	36		4					4	L	40						
	R		40							R		40					
	W			40						M			42	2	16		18
	Y		1	1	38				2	N		1	2	34	3		6
	M					40				NG					60		
	N					2	38		2								
A	TOTAL ERRORS								8								24
D	L	35	1	4					5	L	40						
	R		39			1			1	R		40					
	W			38	1			1	2	M			31	8	21		29
	Y				40					N			2	34	4		6
	M		1			38	1		2	NG				1	58	1	2
	N						40										
D	TOTAL ERRORS								10								37
F	L	33		5	1		1		7	L	39				1		1
	R		37	2				1	3	R		40					
	W			39	1				1	M	1	2	24	1	31	1	36
	Y			1	39				1	N		1	1	33	5		7
	M		1			39			1	NG					60		
	N				1		39		1								
F	TOTAL ERRORS								14								44
G	L	35		3		1		1	5	L	40						
	R		35	4				1	5	R		39		1			1
	W	3		37					3	M		2	38	4	15	1	22
	Y				40					N		1	1	34	3	1	6
	M		1			39			1	NG					59	1	1
	N						40										
G	TOTAL ERRORS								14								30

Table 11:

CONFUSION MATRICES FOR LIST: 7BF

SYSTEM		INITIAL								FINAL							
	S:R	L	R	W	Y	M	N	X		S:R	L	R	M	N	NG	X	
N	L	38	1		1				2	L	37	3					3
	R		38	2					2	R		60					
	W	2		38					2	M	1		51	2	6		9
	Y	1		2	37				3	N				36	4		4
	M	1				39			1	NG			1		39		1
	N					1	39		1								
N	TOTAL ERRORS								11								17
A	L	40								L	40						
	R		38	2					2	R		60					
	W			38	2				2	M			40		20		20
	Y			1	39				1	N				34	6		6
	M	1				39			1	NG					40		
	N					1	39		1								
A	TOTAL ERRORS								7								26
D	L	40								L	37	2				1	3
	R		39		1				1	R		60					
	W			39	1				1	M			49	1	10		11
	Y				40					N				34	6		6
	M	3				37			3	NG			1	4	35		5
	N					1	39		1								
D	TOTAL ERRORS								6								25
F	L	39	1						1	L	38	1		1			2
	R		38	2					2	R		60					
	W	2	2	33	3				7	M			49		11		11
	Y	1		1	38				2	N			2	35	3		5
	M	2	1			37			3	NG			6	4	30		10
	N						40										
F	TOTAL ERRORS								15								28
G	L	40								L	39	1					1
	R		39	1					1	R		59			1		1
	W			39	1				1	M			50	1	9		10
	Y				40					N				37	3		3
	M	1				39			1	NG			1	2	37		3
	N						40										
G	TOTAL ERRORS								3								18

Table 12:

CONFUSION MATRICES FOR LIST: 10M

SYSTEM INITIAL

S:R	S	SL	SW	SM	SN	SP	STSTR	X	
N	S	36		4					4
	SL		40						
	SW			40					
	SM				39			1	1
	SN					40			
	SP						40		
	ST							40	
	STR								
N	TOTAL ERRORS								5
A	S	39		1					1
	SL		40						
	SW			40					
	SM				40				
	SN				1	39			1
	SP						40		
	ST							40	
	STR								
A	TOTAL ERRORS								2
D	S	35		2			3		5
	SL		35	5					5
	SW			40					
	SM				39	1			1
	SN				1	39			1
	SP						40		
	ST							40	
	STR						2	38	2
D	TOTAL ERRORS								14
F	S	34				1		5	6
	SL		28	11		1			12
	SW			40					
	SM			2	36	2			4
	SN				6	34			6
	SP			1			39		1
	ST							39 1	1
	STR			1		1		1 37	3
F	TOTAL ERRORS								33
G	S	38		2					2
	SL		3	37					3
	SW			40					
	SM				40				
	SN					40			
	SP						40		
	ST							40	
	STR								
G	TOTAL ERRORS								5

VII. REFERENCES

1. Nakatani, Lloyd H. and Kathleen D. Dukes, Sensitive Test of Speech Communication Quality. J. Acoust. Soc. Amer., Vol. 53, pp. 1083-1092, 1973.

2. Voiers, William D., Alan D. Sharpley and Carl J. Hehmsoth, Research on Diagnostic Evaluation of Speech Intelligibility. AFCRL-72-0694, September 1972.

APPENDIX A

BBN SPEECH COMPRESSION PROJECT

SUMMARY OF MAJOR RESULTS

1972-1975

NSC Note 77, December 15, 1975

(Author: R. Viswanathan)

BBN SPEECH COMPRESSION PROJECT

SUMMARY OF MAJOR RESULTS

The overall goal of our research has been to develop a Linear Predictive Speech Compression (LPC) system that transmits high quality speech at the lowest possible data rates. We have developed several methods for reducing the redundancy in the speech signal without sacrificing speech quality. Below is a summary of the major results and conclusions of our work in the last three years.

1. Preemphasis

Preemphasis of speech reduces its spectral dynamic range, which in turn (1) diminishes the magnitude of problems due to finite wordlength computation, and (2) improves parameter quantization accuracy. We recommend first-order preemphasis (fixed or adaptive); second-order preemphasis leads to perceivable distortions in synthesized speech [1,2].

2. Variable Order Linear Prediction

We transmit for every frame the minimum number of predictor parameters which adequately represent the speech spectrum in that frame. Our method uses an information theoretic criterion to determine the "optimal" order, and produces average savings of 10% in the transmission rate [2,3].

3. Choice of Parameters for Quantization and Transmission

(a) Pitch: We found that quantizing the logarithm of pitch values was adequate. However, a difficulty arises in attempting to quantize the log pitch in that at the high frequency end (small pitch period) of the range of interest, the quantization bin size, as found by dividing the log pitch scale into equal segments, can be so small as to result in cases where two distinct quantization bins yield the same decoded value, thus wasting some quantization levels. We proposed a method for deriving the pitch coding and decoding tables in such a way that maximum usage is made of the different quantization levels [4].

(b) Gain: Our findings based on statistical error analysis indicated that, in general, it is better to use speech signal energy for transmission than to use prediction error signal energy [5].

(c) Filter Parameters: From a comparative study of a number of equivalent sets of predictor parameters, we

concluded that the reflection coefficients are the best set for transmission purposes. Using a minimax spectral error criterion, we demonstrated that the optimal quantization of the reflection coefficients requires first transforming them to log area ratios (LARs) and then quantizing the LARs linearly [2,6]. Different LARs can be quantized using either the same step size [2,6,7] or different step sizes [8], with the latter resulting in a slight improvement in speech quality over the former.

4. Variable Frame Rate Transmission

LPC parameters are transmitted at variable intervals in accordance with the changing characteristics of the incoming speech. The decision to transmit is based on a threshold on the log likelihood ratio of prediction residuals. We found that, for a given average bit rate, variable frame rate transmission produces superior quality speech than fixed frame rate transmission [2,8,9].

5. Encoding

We use a variable length code (Huffman code) to encode the quantized transmission parameters at significantly lower bit rates (savings on the order of 15%), and with absolutely no effect on speech quality [10].

6. Synthesis

(a) Time-Synchronous Synthesis: We found that time-synchronous updating (e.g., every 5 or 10 msec) of the filter parameters at the synthesizer yields better speech quality than pitch-synchronous updating if the analysis is performed time-synchronously [2]. Time-synchronous parameter updating has the additional advantage of simplifying the necessary computations.

(b) Gain Implementation: We recommend implementing the speech signal energy as a gain multiplier at the input of the synthesizer filter. With the gain multiplier placed at the output of the filter, perceivable distortions are produced in synthesized speech at places where relatively large frame-to-frame energy changes occur [8]. (There are, however, adhoc solutions to this problem.)

(c) Optimal Linear Interpolation: For improved interpolation of synthesizer parameters, we proposed a scheme that requires the transmission of an extra parameter per data frame [11]. This optimal linear interpolation scheme improves speech quality during rapid transitions in the speech signal, at the expense of increasing the bit rate by 50-150 bps.

7. Simulation of LPC Systems

Using floating-point arithmetic we simulated the entire speech compression system with its many different variations in our TENEX time-sharing computer facility [2]. Using this simulation system, we demonstrated the results of three low bit-rate LPC systems at ARPA NSC meetings. The first system produced good quality speech at average rates of 1500 bps[2,12]. Speech quality degraded noticeably for the second system with an average transmission rate of 1000 bps, although the intelligibility of the transmitted speech was still good [8]. The third system, which used differential pulse code modulation (DPCM) for quantizing the transmission parameters, yielded good speech quality at essentially fixed rates of 2000 bps[8]. No explicit silence detection was employed in these three systems.

8. Steps Towards Real-Time Implementation

We worked in cooperation with the other sites in the ARPA community towards implementation of an LPC vocoder that transmits speech in real time over the ARPA Network.

REFERENCES

1. J. Makhoul and R. Viswanathan, "Adaptive Preprocessing for Linear Predictive Speech Compression Systems," presented at the 86th meeting of the Acoust. Soc. Amer., Los Angeles, Oct. 30-Nov. 2, 1973 (also ARPA NSC Note 5).

2. J. Makhoul, R. Viswanathan, L. Cosell and W. Russell, Natural Communication with Computers, Final Report, Vol. II, Speech Compression Research at BBN, Report No. 2976, Dec. 1974.

3. J. Makhoul and C. Cook, "Optimal Number of Poles in a Linear Prediction Model," presented at the 88th meeting of the Acoust. Soc. Amer., St. Louis, Nov. 4-8, 1974.

4. J. Makhoul and L. Cosell, "Recommendations for Encoding and Synthesis," NSC Note 49, Nov. 1974.

5. J. Makhoul and L. Cosell, "Nothing to Lose, but Lots to Gain," NSC Note 56, March 1975.

6. R. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," IEEE Trans. Acoustics, Speech and Signal Processing, Vol. ASSP-23, pp. 309-321, June 1975 (Special issue of papers presented at the Arden House Workshop on Digital Signal Processing, Jan. 1973).

7. R. Viswanathan and W. Russell, "Quantization Routines for Linear Predictive Vocoders," NSC Note 33, July 1974.

8. BBN Quarterly Progress Report on Command and Control Related Computer Technology, Report No. 3093, June 1975.

9. R. Viswanathan and J. Makhoul, "Current Issues in Linear Predictive Speech Compression," Proc. 1974 EASCON Conf., Washington, D.C., pp. 577-585, Oct. 1974.

10. L. Cosell and J. Makhoul, "Variable Wordlength Encoding," NSC Note 34, Aug. 1974 (also presented at the 88th meeting of the Acoust. Soc Amer., St. Louis, Nov. 7-10, 1974).

11. R. Viswanathan, J. Makhoul and W. Russell, "Optimal Interpolation in Linear Predictive Vocoders," BBN Report No. 3065, April 1975 (also presented at the 89th meeting of the Acoust. Soc. Amer., Austin, April 7-11, 975).

12. R. Viswanathan and J. Makhoul, "Towards a Minimally Redundant Linear Predictive Vocoder," presented at the 88th meeting of the Acoust. Soc. Amer., St. Louis, Nov. 7-10, 1974.

ADDITIONAL REFERENCES

1. J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE, Vol. 63, pp. 561-580, April 1975.

2. J. Makhoul, "Spectral Linear Prediction: Properties and Applications," IEEE Trans. Acoustics Speech and Signal Processing, Vol. ASSP-23, pp. 283-296, June 1975.

APPENDIX B

NEW LATTICE METHODS FOR LINEAR PREDICTION

NSC Note 75, December 1, 1975

(Author: John Makhoul)

(This paper will be presented at the 1976 International Conference on Acoustics, Speech and Signal Processing, Philadelphia, April 12-14, 1976.)

NEW LATTICE METHODS FOR LINEAR PREDICTION

This paper presents a new formulation for linear prediction, which we call the covariance lattice method. The method is viewed as one of a class of lattice methods which guarantee the stability of the all-pole filter, with or without windowing of the signal, with finite wordlength computations, and with the number of computations being comparable to the traditional autocorrelation and covariance methods. In addition, quantization of the reflection coefficients can be accomplished within the recursion for retention of accuracy in representation.

1. Introduction

The autocorrelation method of linear prediction [1] guarantees the stability of the all-pole filter, but has the disadvantage that windowing of the signal causes some unwanted distortion in the spectrum. In practice, even the stability is not always guaranteed with finite wordlength (FWL) computations [2]. On the other hand, the covariance method does not guarantee the stability of the filter, even with floating point computation, but has the advantage that there is no windowing of the signal. One solution to these problems was given by Itakura [3] in his lattice

formulation. In this method, filter stability is guaranteed, with no windowing, and with FWL computations. Unfortunately, this is accomplished with about a four-fold increase in computation over the other two methods.

This paper presents a class of lattice methods which have all the properties of a regular lattice but where the number of computations is comparable to the autocorrelation and covariance methods. In these methods the "forward" and "backward" residuals are not computed. The reflection coefficients are computed directly from the covariance of the input signal.

2. Lattice Formulations

In linear prediction, the signal spectrum is modelled by an all-pole spectrum with a transfer function given by

$$H(z) = \frac{G}{A(z)} , \tag{1}$$

$$\text{where} \quad A(z) = \sum_{k=0}^{p} a_k z^{-k} , \; a_0 = 1 , \tag{2}$$

is known as the inverse filter, G is a gain factor, a_k are the predictor coefficients, and p is the number of poles or predictor coefficients in the model. If H(z) is stable, A(z) can be implemented as a lattice filter, as shown in Fig. 1. The reflection (or partial correlation) coefficients Ki in the lattice are uniquely related to the predictor coefficients. Given Ki, $1 \le i \le p$, the set {ak} is

computed by the recursive relation:

$$a_i^{(i)} = K_i$$
$$a_j^{(i)} = a_j^{(i-1)} + K_i\, a_{i-j}^{(i-1)} , \quad 1 \le j \le i-1, \tag{3}$$

where the equations in (3) are computed recursively for $i=1,2,\ldots,p$. The final solution is given by $a_j=a_j^{(p)}$, $1\le j\le p$. For a stable $H(z)$, one must have:

$$|K_i| < 1, \quad 1\le i\le p . \tag{4}$$

In the lattice formulation, the reflection coefficients can be computed by minimizing some error norm of the forward residual $f_m(n)$ or the backward residual $b_m(n)$, or a combination of the two. From Fig. 1, the following relations hold:

$$f_0(n) = b_0(n) = s(n) , \tag{5a}$$
$$f_{m+1}(n) = f_m(n) + K_{m+1}\, b_m(n-1) , \tag{5b}$$
$$b_{m+1}(n) = K_{m+1}\, f_m(n) + b_m(n-1) . \tag{5c}$$

$s(n)$ is the input signal and $e(n)=f_p(n)$ is the output residual.

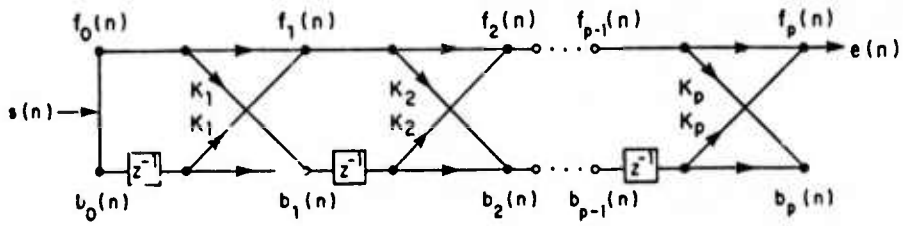


Fig. 1. Lattice inverse filter.

We shall give several methods for the determination of the reflection coefficients. These methods depend on different ways of correlating the forward and backward residuals. Below, we shall make use of the following definitions:

$$F_m(n) = E[f_m^2(n)] \tag{6a}$$

$$B_m(n) = E[b_m^2(n)] \tag{6b}$$

$$C_m(n) = E[f_m(n)b_m(n-1)] \ , \tag{6c}$$

where $E(.)$ denotes expected value. The left hand side of each of the equations in (6) is a function of n because we are making the general assumption that the signals are nonstationary. (Subscripts, etc., will be dropped sometimes for convenience.)

(a) Forward Method

In this method the reflection coefficient at stage m+1 is obtained as a result of the minimization of an error norm given by the variance (or mean square) of the forward residual:

$$F_{m+1}(n) = E[f_{m+1}^2(n)] \ . \tag{7}$$

By substituting (5b) in (7) and differentiating with respect to Km+1, one obtains:

$$K_{m+1}^{f} = - \frac{E[f_m(n) b_m(n-1)]}{E[b_m^2(n-1)]} = - \frac{C_m(n)}{B_m(n-1)} \qquad (8)$$

This method of computing the filter parameters is similar to the autocorrelation and covariance methods in that the mean squared forward error is minimized.

(b) Backward Method

In this case, the minimization is performed on the variance of the backward residual at stage m+1. From (5c) and (6b), the minimization of $B_{m+1}(n)$ leads to:

$$K_{m+1}^{b} = - \frac{E[f_m(n) b_m(n-1)]}{E[f_m^2(n)]} = - \frac{C_m(n)}{F_m(n)} \quad . \qquad (9)$$

Note that, since $F_m(n)$ and $B_m(n-1)$ are both nonnegative and the numerators in (8) and (9) are identical, K^f and K^b always have the same sign S:

$$S = \text{sign } K^f = \text{sign } K^b \quad . \qquad (10)$$

(c) Geometric Mean Method (Itakura)

The main problem in the above two techniques is that the computed reflection coefficients are not always guaranteed to be less than 1 in magnitude, i.e., the stability of H(z) is not guaranteed. One solution to this problem was offered by Itakura [3] where the reflection coefficients are computed from

$$K^I_{m+1} = -\frac{E[f_m(n)b_m(n-1)]}{\sqrt{E[f_m^2(n)]E[b_m^2(n-1)]}} \qquad (11)$$

$$= -\frac{C_m(n)}{\sqrt{F_m(n)B_m(n-1)}} \quad .$$

K^I_{m+1} is the negative of the statistical correlation between $f_m(n)$ and $b_m(n-1)$; hence, property (4) follows. To the author's knowledge, (11) cannot be derived directly by minimizing some error criterion. However, from (8), (9) and (11), one can easily show that K^I is the geometric mean of K^f and K^b:

$$K^I = S\sqrt{K^f K^b} \qquad (12)$$

where S is given by (10). From the properties of the geometric mean, it follows that:

$$\min[|K^f|,|K^b|] \le |K^I| \le \max[|K^f|,|K^b|] \quad . \qquad (13)$$

Now, since $|K^I|<1$, it follows that <u>if the magnitude of</u>

either K^f or K^b is greater than 1, the magnitude of the other is necessarily less than 1. This leads us to another definition for the reflection coefficient.

(d) Minimum Method

$$K^M = S \min[|K^f|, |K^b|] \quad . \tag{14}$$

This says that, at each stage, compute K^f and K^b and choose as the reflection coefficient the one with the smaller magnitude.

(e) General Method

Between K^M and K^I there are an infinity of values that can be chosen as valid reflection coefficients (i.e., $|K|<1$). These can be conveniently defined by taking the generalized rth mean of K^f and K^b:

$$K^r = S \left[\frac{1}{2}(|K^f|^r + |K^b|^r) \right]^{1/r} \quad . \tag{15}$$

As $r \to 0$, $K^r \to K^I$, the geometric mean. For $r>0$, K^r cannot be guaranteed to satisfy (4). Therefore, for K^r to be a reflection coefficient, we must have $r \leq 0$. In particular:

$$K^0 = K^I, \; K^{-\infty} = K^M \quad . \tag{16}$$

If the signal is stationary, one can show that $K^f = K^b$, and that

$$K^r = K^f = K^b, \text{ all } r. \text{ (Stationary Case)} \tag{17}$$

(f) Harmonic Mean Method (Burg)

There is one value of r for which K^r has some interesting properties, and that is r=-1. K^{-1}, then, would be the harmonic mean of K^f and K^b :

$$K^B = K^{-1} = \frac{2K^fK^b}{K^f+K^b} = -\frac{2C_m(n)}{F_m(n)+B_m(n-1)} \quad . \tag{18}$$

One can show that

$$|K^M| \leq |K^B| \leq |K^I| \quad . \tag{19}$$

In fact, Itakura used K^B as an approximation to K^I in (11) to avoid computing the square root.

One important property of K^B that is not shared by K^I and K^M, is that K^B results directly from the minimization of an error criterion. The error is defined as the sum of the variances of the forward and backward residuals:

$$E_{m+1}(n) = F_{m+1}(n) + B_{m+1}(n) \quad . \tag{20}$$

Using (5) and (6), one can show that the minimization of (20) indeed leads to (18). One can also show that the forward and backward minimum errors at stage m+1 are related to those at stage m by the following:

$$F_{m+1}(n) = \left[1-(K^B_{m+1})^2\right] F_m(n) \tag{21a}$$

$$B_{m+1}(n) = \left[1-(K^B_{m+1})^2\right] B_m(n-1) \quad . \tag{21b}$$

This formulation is originally due to Burg [4]; it has been used recently by Boll [5] and Atal [6].

(g) Discussion

If the signal s(n) is stationary, all the methods described above give the same result. In general, the signal cannot be assumed to be stationary and the different methods will give different results. Which method to choose in a particular situation is not clear cut. We tend to prefer the use of K^B in (18) because it minimizes a reasonable and well defined error and guarantees stability simultaneously, even for a nonstationary signal.

3. The Covariance-Lattice Method

If linear predictive analysis is to be performed on a regular computer, the number of computations for the lattice methods given above far exceeds that of the autocorrelation and covariance methods (see the first row of Fig. 2). This is unfortunate since, otherwise, lattice methods have superior properties when compared to the autocorrelation and covariance methods (see Fig. 3). Below, we derive a new method, called the covariance-lattice method, which has all the advantages of a regular lattice, but with an efficiency comparable to the two non-lattice methods.

	AUTOCORRELATION METHOD	COVARIANCE METHOD	REGULAR LATTICE (WITH RESIDUALS)
TRADITIONAL METHODS	$pN + p^2$	$pN + \frac{1}{6}p^3 + \frac{3}{2}p^2$	$5\,pN$
NEW LATTICE METHODS	$pN + \frac{1}{6}p^3 + \frac{3}{2}p^2$	$pN + \frac{1}{2}p^3 + 2p^2$	$5\,pN$

Fig. 2. Computational cost for traditional as compared to new lattice methods.

LINEAR PREDICTION METHOD	ADVANTAGES	DISADVANTAGES
AUTOCORRELATION	1. THEORETICAL STABILITY 2. COMPUTATIONALLY EFFICIENT	1. WINDOWING 2. POSSIBLE INSTABILITY WITH FWL COMPUTATION
COVARIANCE	1. NO WINDOWING 2. COMPUTATIONALLY EFFICIENT	1. STABILITY NOT GUARANTEED EVEN WITH FLOATING POINT
REGULAR LATTICE	1. WINDOWING NOT NECESSARY 2. STABILITY CAN BE GUARANTEED 3. NUMBER OF SAMPLES FOR ANALYSIS CAN BE REDUCED 4. REFLECTION COEFFICIENTS CAN BE QUANTIZED WITHIN RECURSION	1. COMPUTATIONALLY EXPENSIVE
COVARIANCE LATTICE	1-4. SAME AS FOR REGULAR LATTICE METHOD 5. COMPUTATIONALLY EFFICIENT	

Fig. 3. Comparison between different LP methods.

From the recursive relations in (3) and (5), one can show that

$$f_m(n) = \sum_{k=0}^{m} a_k^{(m)} s(n-k) \,, \tag{22a}$$

$$b_m(n) = \sum_{k=0}^{m} a_k^{(m)} s(n-m+k) \,. \tag{22b}$$

Squaring (22a) and taking the expected value, there results

$$F_m(n) = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k^{(m)} a_i^{(m)} \phi(k,i) \,, \tag{23}$$

$$\text{where } \phi(k,i) = E\,[s(n-k)s(n-i)] \tag{24}$$

is the nonstationary autocorrelation (or covariance) of the signal s(n). ($\phi(k,i)$ in (24) is technically a function of n, which has been dropped for convenience.) In a similar fashion one can show from (22b), with n replaced by n-1, that

$$B_m(n-1) = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k^{(m)} a_i^{(m)} \phi(m+1-k, m+1-i) \,, \tag{25}$$

$$C_m(n) = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k^{(m)} a_i^{(m)} \phi(k, m+1-i) \,. \tag{26}$$

Given the covariance of the signal, the reflection coefficient at stage m+1 can be computed from (23), (25) and (26) by substituting them in the desired formula for K_{m+1}. The name "covariance-lattice" stems from the fact that this is basically a lattice method that is computed from the covariance of the signal; it can be viewed as a way of stabilizing the covariance method. One salient feature is

that the forward and backward residuals are never actually computed in this method. But this is not different from the non-lattice methods.

In the harmonic mean method (18), $F_m(n)$ need not be computed from (23); one can use (21a) instead, with m replaced by m-1. However, one must use (25) to compute $B_m(n-1)$; (21b) cannot be used because $B_{m-1}(n-2)$ would be needed and it is not readily available.

(a) Stationary Case

For a stationary signal, the covariance reduces to the autocorrelation:

$$\phi(k,i) = R(i-k) = R(k-i). \quad \text{(Stationary)} \tag{27}$$

From (23-27), it is clear that

$$F_m = B_m = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k^{(m)} a_i^{(m)} R(i-k) \ , \tag{28}$$

$$\text{and } C_m = \sum_{k=0}^{m} \sum_{i=0}^{m} a_k^{(m)} a_i^{(m)} R(m+1-i-k) \ . \tag{29}$$

Making use of the normal equations [1]

$$\sum_{i=0}^{m} a_i^{(m)} R(i-k) = 0, \quad 1 \le k \le m \ , \tag{30}$$

and of (21), one can show that the stationary reflection coefficient is given by:

$$K_{m+1} = -\frac{C_m}{F_m} = -\frac{\sum_{k=0}^{m} a_k^{(m)} R(m+1-k)}{(1-K_m^2)F_{m-1}} \qquad (31)$$

with $F_0=R_0$. (31) is exactly the equation used in the autocorrelation method.

(b) Quantization of Reflection Coefficients

One of the features of lattice methods is that the quantization of the reflection coefficients can be accomplished within the recursion, i.e., K_m can be quantized before K_{m+1} is computed. In this manner, it is hoped that some of the effects of quantization can be compensated for.

In applying the covariance-lattice procedure to the harmonic mean method, one must be careful to use (23) and not (21a) to compute $F_m(n)$. The reason is that (21a) is based on the optimality of K^B, which would no longer be true after quantization.

Similar reasoning can be applied to the autocorrelation method. Those who have tried to quantize K_m inside the recursion, have no doubt been met with serious difficulties. The reason is that (31) assumes the optimality of the predictor coefficients at stage m, which no longer would be true if K_m were quantized. The solution is to use (28) and (29), which make no assumptions of optimality. Thus, we

have what we shall call the autocorrelation lattice method, where there is only one definition of K_{m+1}:

$$K_{m+1} = -\frac{C_m}{F_m}, \quad \text{(Autocorrelation-Lattice)} \tag{32}$$

where F_m and C_m are given by (28) and (29).

4. Computational Issues

(a) Simplifications

Equations (23),(25) and (26) can be rewritten to reduce the number of computations by about one half. The results for $C_m(n)$ and $F_m(n)+B_m(n-1)$ can be shown to be as follows:

$$\begin{aligned} C_m(n) = {} & \phi(0,m+1) + \sum_{k=1}^{m} a_k^{(m)} [\phi(0,m+1-k)+\phi(k,m+1)] \\ & + \sum_{k=1}^{m} [a_k^{(m)}]^2 \phi(k,m+1-k) \\ & + \sum_{k=1}^{m-1} \sum_{i=k+1}^{m} a_k^{(m)} a_i^{(m)} [\phi(k,m+1-i)+\phi(i,m+1-k)] \end{aligned} \tag{33}$$

$$\begin{aligned} F_m(n)+B_m(n-1) = {} & \phi(0,0)+\phi(m+1) \\ & + 2\sum_{k=1}^{m} a_k^{(m)} [\phi(0,k)+\phi(m+1,m+1-k)] \\ & + \sum_{k=1}^{m} [a_k^{(m)}]^2 [\phi(k,k)+\phi(m+1-k,m+1-k)] \\ & + 2\sum_{k=1}^{m-1} \sum_{i=k+1}^{m} a_k^{(m)} a_i^{(m)} [\phi(k,i)+\phi(m+1-k,m+1-i)] \end{aligned} \tag{34}$$

(28) and (29) can also be simplified in a similar fashion.

(b) Covariance Computation

If the signal is known for $0 \leq n \leq N-1$, then one common method to compute the covariance is

$$\phi(k,i) = \sum_{n=p}^{N-1} s(n-k)s(n-i) \quad , \tag{35}$$

where p is the order of the desired predictor.

(c) Computational Cost

Fig. 2 shows a comparison of the number of computations for the different methods, where terms of order p have been neglected. The increase in computation for the covariance lattice method over non-lattice methods is not significant if N is large compared to p, which is usually the case. Furthermore, in the covariance lattice method, the number of signal samples can be reduced to about half that used in the autocorrelation method. This, not only reduces the number of computations, but also improves the spectral representation by reducing the amount of averaging.

5. Procedure

Below is the complete algorithm for what we believe currently to be the best overall method for linear predictive analysis. It comprises the harmonic mean

definition (18) for the reflection coefficients, and the covariance lattice method.

(a) Compute the covariances $\phi(k,i)$ for $k,i=0,1,\ldots,p$.

(b) $m \leftarrow 0$.

(c) Compute $C_m(n)$ and $F_m(n)+B_m(n-1)$ from (33) and (34), or from (23),(25) and (26).

(d) Compute K_{m+1} from (18).

(e) Quantize K_{m+1} if desired (perhaps using log area ratios [7] or some other technique).

(f) Using (3), compute the predictor coefficients $\{a_k^{(m+1)}\}$ from $\{a_k^{(m)}\}$ and K_{m+1}. Use the quantized value if K_{m+1} was quantized in (d).

(g) $m \leftarrow m+1$.

(h) If $m<p$, go to (c); otherwise exit.

References

[1] J. Makhoul, "Linear Prediction: A Tutorial Review," Proc. IEEE, 561-580, April 1975.

[2] J. Markel and A. Gray, Jr., "Fixed-Point Truncation Arithmetic Implementation of a Linear Prediction Autocorrelation Vocoder," IEEE Trans. ASSP, 273-281, 1974.

[3] F. Itakura and S. Saito, "Digital Filtering Techniques for Speech Analysis and Synthesis," 7th Int. Cong. Acoust., Budapest, 25-C-1, 1971.

[4] J. Burg, "A New Analysis Technique for Time Series Data," NATO Advanced Study Institute on Signal Processing, Enschede, Netherlands, 1968.

[5] S. Boll, "Selected Methods for Improving Synthesis Speech Quality Using Linear Predictive Coding: System Description, Coefficient Smoothing and STREAK," UTEC-CSc-74-151, Comp. Science Dept., Univ. Utah, 1974.

[6] B. Atal, M. Schroeder and V. Stover, "Voice-Excited

Predictive Coding System for Low Bit-Rate Transmission of Speech," Int. Conf. Comm., San Francisco, June 1975.

[7] R. Viswanathan and J. Makhoul, "Quantization Properties of Transmission Parameters in Linear Predictive Systems," IEEE Trans. ASSP, 309-321, June 1975.

APPENDIX C

SPECIFICATIONS FOR ARPA-LFC SYSTEM II

NSC Note 82, February 12, 1976

(Authors: R. Viswanathan and John Makhoul)

I. INTRODUCTION

This note provides specifications for ARPA-LPC speech compression system II, an update of the present system I. The approach we employed in arriving at these specifications has been to reap maximum benefit for the least amount of effort. Our overall design objective has been to achieve average continuous-speech transmission rates of about 2200 bps. With the use of a silence detection algorithm, these rates may be expected to drop to about 1000 bps or less.

The following sections deal with only those aspects of System I which need to be modified. The major differences between Systems I and II are due to:

1. Variable frame rate (VFR) transmission, and
2. New coding/decoding tables for transmission parameters.

Compared to System I, VFR transmission should yield a lower (average) frame rate, while new coding/decoding tables employ fewer bits per transmitted frame. Thus, both modifications contribute in lowering the average bit rate.

The specific recommendations put forth in this note represent a first cut on our part. Comments and suggestions are welcome.

In the preparation of this note we have had discussions about implementation of VFR transmission on the SPS-41 with

Earl Craighill, Danny Cohen and Lynn Cosell. Joe Tierney supplied the statistics for the reflection coefficients. Randy Cole explained the details of the present gain table.

II. PARCELS, PACKETS AND NEGOTIATIONS

Since the discussion of the proposed VFR transmission scheme requires the knowledge of the particular parcel format chosen, we first consider the latter issue in this section along with related issues, such as packet format and negotiations to establish a voice link on the ARPANET.

A. Parcel Format

We propose that a variable-length parcel be transmitted for every analysis frame. A parcel has a 3-bit header: first bit is 1, only if the parcel contains pitch data; similarly, second and third header bits indicate if the parcel contains codes of gain and reflection coefficients (or k-parameters), respectively, A parcel, therefore, can be as small as 3 bits, and as large as 50 bits. (The breakdown of the possible additional 47 bits among pitch, gain and reflection coefficients is given in Section IV.)

The parcel format just described allows the use of a separate transmission criterion for each of the three groups of analysis parameters: pitch, gain, and reflection coefficients. The primary reasons for proposing this

independent transmission policy are: 1) It is the most general approach, and therefore individual variations can be implemented with relative ease. 2) In general, significant variations in each of the three parameter groups do not occur simultaneously. Our experience with low average frame-rate transmission has shown that if pitch and gain are transmitted only when reflection coefficients are transmitted, perceivable speech quality distortions result [1].

We have considered an alternate parcel format whereby a parcel of data is transmitted, not for every analysis frame, but only when a parameter transmission occurs. This means that the parcel should also contain a code to specify the interval between transmissions, which is variable on account of VFR transmission. The disadvantages of this alternate format are as follows. First, the maximum transmission interval has to be restricted to be small so it can be coded using a small number of bits. For example, a code length of 3 bits means that the transmission interval can only be as long as 8 analysis frames. Secondly, independent transmission of pitch, gain and reflection coefficients requires the transmission of 3 separate codes corresponding to the 3 independent transmission intervals. For the range of average frame rates we are interested in, the resulting parcel overhead is more than the overhead required by the proposed parcel format. These reasons justify our choice of

the simple 3-bit-headered parcel format for use in System II.

B. Packet Format

The packet header details are the same as discussed in [2]. With VFR transmission, we suggest the use of a variable-length packet whereby the transmission delay (or packet loading time) is limited. Our recommendation is to limit the packet size such that the packet loading time is less than, say, 400 msec. In other words, a packet is transmitted either when it is fully loaded with an integer number of parcels, or when the total speech duration it represents is about 400 msec, whichever happens first.

Since the proposed parcel format does not restrict the interval between two successive parameter transmissions, it can happen that a packet is full of parcels having header bits only (i.e., no parcel has parameter data in it). This event happens usually for long pauses or silence. If the silence duration exceeds 1 sec, the silence detection algorithm steps in to send a silence packet. If the duration is less than 1 sec, it is possible to have even two successive packets containing header-only parcels. This poses a problem if the receiver performs parameter interpolation between transmissions inasmuch as the receiver has to buffer two or more packets, thus producing a large

reconstitution delay. We have thought of a number of solutions to this problem, such as forcing a packet to have at least one data parcel. The following solution seems to be the most reasonable one. When a parameter transmission interval exceeds, say, 100 msec, then the last transmitted parameter values are used for the duration. (The value, 100 msec, is given here only as a guide. Other reasonable values may be used.) Thus, when a long transmission interval (less than 1 sec) is encountered, this method repeats the last transmitted data for all analysis frames in the interval, except the last stretch of less than 100 msec duration for which interpolation is performed to generate the parameter data.

C. Negotiations

We suggest an update of the present NVP program to include the various <WHAT> and <HOW> negotiations given on pp. 6-7 in [2]. This recommendation calls for parameterization of analysis and synthesis programs in terms of variables such as sample period, LPC order, and samples per parcel (or interframe interval, IFI). For sample period = 150 microseconds, IFI may be either 9.6 msec (64 samples) or 19.2 msec (128 samples). The coding/decoding tables given in Section IV constitute table-set 2 for the negotiation item 10 on p. 7 in [2].

D. System I: A Special Case of System II

The discussions presented above clearly show that the present fixed rate LPC System I can be viewed as a special case of System II upon selection of the negotiable parameter values to be as those for Version 1 (p. 7, [2]). The only difference is that the transmission bit rate will be increased by 52 x 3 = 156 bps due to the 3-bit/parcel overhead. Thus, after implementing System II, we recommend running it in System I mode as an initial debugging test.

III. VARIABLE FRAME RATE TRANSMISSION

The idea of VFR transmission has been well explored both at SRI [3] and at BBN [4]. Since these references contain detailed discussions about the VFR scheme, we provide below only those details relevant to System II implementation. First, however, some general comments are in order.

A number of criteria (or distance measures) may be used in deciding when to transmit LPC parameters, i.e., in deciding if the parameters have changed sufficiently to warrant a new transmission. Fortunately, different LPC implementations (or sites) can use different criteria but still preserve compatibility to communicate with each other. This means that no negotiation is needed regarding the transmission criterion, and more importantly, one can

experiment with different transmission criteria by changing the transmitter program only, without having to worry about the receiver programs located either locally (back-to-back mode) or remotely.

As mentioned in Section II, we recommend the use of separate transmission criteria for pitch, gain and reflection coefficients. Below we present previously tested transmission criteria for reflection coefficients, and mention possibilities that are being currently investigated for pitch and gain.

A. Reflection Coefficients

We shall consider a specific transmission criterion for reflection coefficients. This is the so-called likelihood ratio or ratio of prediction residual energies [3-5]. This VFR scheme transmits the reflection coefficients of a given analysis frame only if the likelihood ratio computed between that frame and the last transmitted frame exceeds a threshold, denoted by LRT (likelihood ratio threshold).

To compute the likelihood ratio, we need to compute for each analysis frame the autocorrelations $\{b_i\}$ of the predictor coefficients $\{a_i\}$:

$$b_i = \sum_{j=0}^{M-i} a_j a_{j+i} \quad , \quad a_0 = 1 \quad , \quad 0 \leq i \leq M \quad ,$$

where M is the predictor order. The analysis program should compute these M+1 autocorrelations and transfer them along with the already available preemphasized speech autocorrelations $\{R_i\}$ and minimum residual energy α_M to the transmitter program containing the VFR scheme.

Below is a step-by-step procedure of the VFR transmission scheme. The superscript n used with the quantities b_i, R_i and α_M denotes their values corresponding to the n-th analysis frame.

(1) Transmit coefficients of frame n

$$b_j \longleftarrow b_j^{(n)} , \quad 0 \leq j \leq M$$

$$i \longleftarrow 0 .$$

(2) $i \longleftarrow i + 1$

$$R_j \longleftarrow R_j^{(n+i)} , \quad 0 \leq j \leq M$$

$$\alpha_M \longleftarrow \alpha_M^{(n+i)}$$

$$D \longleftarrow b_0 R_0 + 2 \sum_{j=1}^{M} b_j R_j - \alpha_M \, LRT$$

(3) If $D \leq 0$, go to (2). (No transmission).

(4) $n \longleftarrow n + i$, go to (1).

We suggest a value of LRT=1.4 for System II.

Earl Craighill has told us about an approximation (originally suggested by Steve Boll) to the likelihood ratio in terms of reflection coefficients of appropriate analysis frames. Since the performance of this approximation has not been well studied and, more importantly, since the direct computation given above is, according to Danny Cohen, within the time constraints of existing real-time implementations, we have not presented the details of the approximation.

Other Suggestions

We have investigated two modifications of the above basic likelihood ratio method in the context of developing a 1000 bps LPC system [1]. These may be used in System II to improve speech quality primarily.

1. The first modification is to use a slightly higher threshold (about 5-10% higher*) for unvoiced sounds than for voiced sounds. When a transmission interval contains a transition between voiced and unvoiced sounds, the lower threshold is always employed to encourage a transmission.
2. The second modification involves the use of a double

---

*These percentage figures are different from those given in [1] because there we used logarithm of the likelihood ratio in the transmission criterion.

threshold strategy. Two likelihood ratio thresholds, LRT1 and LRT2, are employed in this scheme. LRT2 may be about 20% higher* than LRT1 (e.g. LRT1=1.4 and LRT2=1.7). The idea behind this modification is that if the likelihood ratio between a current frame and the previously transmitted frame exceeds only LRT1, and not LRT2, then the current frame is transmitted; if it exceeds both thresholds, then the frame immediately preceding the current frame is transmitted. The latter step avoids having to do parameter interpolation between largely different data frames. A step-by-step procedure of the modified scheme is given in the next page.

---

*See footnote on page 9.

(1) Transmit coefficients of frame n

$b_j \leftarrow b_j^{(n)}$, $0 \leq j \leq M$

$i \leftarrow 0$.

(2) $i \leftarrow i + 1$

$R_j \leftarrow R_j^{(n+i)}$, $0 \leq j \leq M$

$\alpha_M \leftarrow \alpha_M^{(n+i)}$.

$D1 \leftarrow b_0 R_0 + 2 \sum_{j=1}^{M} b_j R_j - \alpha_M$ LRT1.

(3) If $D1 \leq 0$, go to (2). (No transmission).

(4) If $i = 1$, go to (7).

(5) $D2 \leftarrow D1 - \alpha_M$ (LRT2 - LRT1)

If $D2 \leq 0$, go to (7).

(6) $i \leftarrow i - 1$.

(7) $n \leftarrow n + i$, go to (1).

As a first step, we recommend implementing the basic likelihood ratio method. Later, one may want to try out some variations. such as the ones discussed above. Such experimentation may be facilitated by having the transmitter program reside in a computer that allows the program changes to be done relatively easily (e.g. PDP-11 rather than SPS-41).

B. Pitch and Gain

Currently, we are investigating transmission criteria (separate for pitch and gain) which transmit the parameter if it has changed by more than a prespecified amount since the last transmission. We will report the results of this work in a later NSC note. The step-by-step description of a typical scheme is given below, where T denotes a preselected threshold. (A double threshold strategy may also be used here as well.)

(1) Transmit value at frame n

$i \leftarrow 0$.

(2) $i \leftarrow i + 1$

$D \leftarrow |(\text{frame } n+i \text{ value}) - (\text{frame } n \text{ value})| - T$

(3) If $D \leq 0$, go to (2). (No transmission).

(4) $n \leftarrow n + i$, go to (1).

For now, we recommend implementing the simple method of transmitting gain at a fixed rate of every 19.2 msec, and pitch also at the same fixed rate except during an unvoiced region where only the pitch value (=0) of the first unvoiced frame is transmitted; the receiver continues the unvoiced status until a new pitch value is received.

IV. CODING/DECODING TABLES

For System II, we recommend the use of a new set of coding/decoding tables for transmission parameters. The gain table in the new set is the same as that given in NSC Note 68 [2] except for a suggestion of using a nonzero decoded value for the zero level. The pitch table has been designed in such a way that decoded values are unique (or unequal) thus employing the available quantization levels more efficiently [6]. Tables for reflection coefficients, on the other hand, have been designed to employ fewer total number of bits than what the tables of System I require. The resulting bit savings (about 20 bits/transmitted frame) are due to: 1) the use of smaller parameter ranges obtained from real speech data, 2) the efficient selection of step sizes for the different parameters (log area ratios or LARs) based on the spectral sensitivity concept [1], and 3) the LPC order M being 9 instead of 10. As an important consequence, a different table is proposed for each reflection coefficient [1].

A. Bit Allocation

The new quantization tables given below are based on the following bit allocation: pitch = 6 bits; gain = 5 bits; 9 reflection coefficients k(1) to k(9), in that order = 5, 5, 5, 4, 4, 4, 3, 3, 3 bits. Thus, a transmitted frame of data (parcel) has a maximum of 47 data bits (plus 3 header bits).

Our feeling is that a 9-th order LPC analysis is adequate for a sampling rate of 6.7 kHz. However, if one wants to have M=10, we suggest duplicating the coding/decoding table of the 9-th coefficient to be used for the 10-th.

B. General Comments About Quantization Tables

Pitch and gain tables given in the following pages are arranged in three columns "X(J)", "J" and "R(J)", while the tables for the reflection coefficients have two additional columns "INDEX(J)" and "INDEXP(J)". (These two columns are explained later.) Notice that the entries in the first column "X(J") are half a step off the other columns. This is to indicate that intervals from the X-domain (pitch, gain, and the reflection coefficients) are mapped into codes or levels "J", which are transmitted over the network, to be translated by the receiver into the values in the column "R(J)". These intervals are open-close intervals as defined

in [2]. Values of a parameter above and below the range of the "X(J)" column are mapped into the maximum and minimum entries of the "J" column.

C. Pitch Table

The pitch table given here is the "optimal" solution presented in NSC Note 49 [6]. Briefly, the logarithm of the pitch period in number of samples was quantized. A difficulty arises in attempting to quantize the log pitch in that at the high frequency end (small pitch period) of the range of interest, the quantization bin size, as found by dividing the log pitch scale into equal segments, can be so small as to result in cases where two distinct quantization bins yield the same decoded value, thus wasting some quantization levels. We used a method, for deriving the pitch coding and decoding tables, which ensures maximum usage of all the available quantization levels [6].

The scaling of the pitch value obtained from SIFT program is the same as before. (Scale up by shifting 9 places to the left, i.e., multiplying by 512. Since NSC Note 42 has not been issued yet, the only reference for this scaling seems to be NSC Note 36 [7].)

The level J=0 defines the unvoiced condition. The receiver decodes it as the interframe interval (IFI) expressed in number of samples. As we recommended in

NEW PITCH TABLE

X(J)	J	R(J)	X(J)	J	R(J)	X(J)	J	R(J)
0			7254			12031		
	0	64*		22	40		43	68
0			7424			12265		
	1	19		23	41		44	70
3840			7595			12636		
	2	20		24	42		45	72
4011			7764			12969		
	3	21		25	43		46	74
4182			7942			13313		
	4	22		26	44		47	76
4352			8085			13654		
	5	23		27	45		48	78
4523			8362			13995		
	6	24		28	47		49	80
4694			8641			14336		
	7	25		29	48		50	82
4864			8789			14678		
	8	26		30	49		51	84
5035			8940			15018		
	9	27		31	50		52	86
5206			9213			15366		
	10	28		32	52		53	88
5376			9502			15680		
	11	29		33	53		54	90
5547			9613			16126		
	12	30		34	54		55	93
5718			9906			16583		
	13	31		35	56		56	95
5888			10154			16874		
	14	32		36	57		57	97
6059			10410			17301		
	15	33		37	59		58	100
6230			10669			17862		
	16	34		38	60		59	103
6400			10919			18261		
	17	35		39	62		60	105
6571			11188			18667		
	18	36		40	63		61	108
6742			11404			19201		
	19	37		41	65		62	111
6912			11806			19733		
	20	38		42	67		63	114
7083			12031			Infinity		
	21	39						
7254								

*This value is the interframe interval in number of samples.

Section II, IFI is a variable whose value is decided at the time of the negotiations. The pitch table gives a decoded value of 64 for J=0, assuming IFI=9.6 msec. For any other value of IFI, this decoded value has to be changed.

D. Gain Table

This is the same gain table as given in NSC Note 68 [2]. The "X(J)" column is the square root of the energy (or the zero-lag autocorrelation R_0) of the preemphasized and windowed speech signal. The gain table assumes a maximum X-value of 3000 and allows for a dynamic range of about 43.5 dB. (With a 12-bit A/D input (including the sign bit) and with 128 samples in the analysis interval, R_0 is assumed to have a maximum value of about 2^{23} after accounting for a 6 dB (1 bit) difference between peak and rms values of speech [7] and a combined loss of about 12 dB (2 bits) due to preemphasis and windowing. Notice that $\sqrt{2^{23}}$ is about 3000. These numbers were supplied to us by Randy Cole. Since they are not given in [2], we have included them in this note.)

Our experience has shown that using R_0=0 for the zeroth level can cause perceivable problems in the synthesized speec [1]. These problems arise due to: 1) certain very low energy speech sections (e.g. beginnings of [h], [n], [d]) being somewhat cutoff in the synthesized version, and

GAIN ($\sqrt{R_0}$) TABLE

(Taken from NSC Note 68)

X(J)	J	R(J)	X(J)	J	R(J)
0			225		
	0	0*		16	245
20			266		
	1	20		17	289
22			315		
	2	24		18	342
26			372		
	3	28		19	404
30			439		
	4	33		20	478
36			519		
	5	39		21	565
42			614		
	6	46		22	667
50			725		
	7	54		23	789
59			857		
	8	64		24	932
70			1013		
	9	76		25	1101
83			1197		
	10	90		26	1301
98			1415		
	11	106		27	1538
116			1672		
	12	126		28	1818
137			1976		
	13	148		29	2148
161			2335		
	14	175		30	2539
191			2760		
	15	207		31	3000
225			Infinity		

*We recommend the use of a nonzero number such as 15(-46dB) or 10 (-50dB) for this decoded value.

2) having to listen to the contrast between absolute silence and the usually noisy synthesized speech. These problems generally disappear if we use a relatively small nonzero energy for the level J=0. Therefore, we recommend decoding this level as a small value such as 15 (about 46 dB lower than the maximum value of 3000) or 10 (about 50 dB lower than the maximum).

E. Tables for Reflection Coefficients

The 9 coding/decoding tables given, one for each coefficient, represent linear quantization of log area ratios with a different step size for each coefficient [1]. The scaling of the transmitter table values is the same as in [2]. In other words, the "X(J)" column of the table for the i-th reflection coefficient k_i has entries of the form $k_i 2^{15}$. The receiver table "R(J)" gives the decoded values of the reflection coefficients in the same scaled form. The column "INDEX(J)" gives the indices into the SPS sine table corresponding to the decoded values i.e., these entries are of the form $\arcsin(k_i)\ 2^{15}/\pi$. These entries refer to the "fine" SPS sine table, which calls for additional multiplications, thus increasing the computational time. The entries in the "INDEXP(J)" column, on the other hand, are indices into the "coarse" sine table only, thus requiring no such multiplications; these indices, being integer multiples of 2^7, are the closest approximations to

the corresponding ones in the "INDEX(J)" column. (It is important to note that we have factored 2^7 out of the entries in the last column.)

As mentioned in the beginning of this section, in deriving these tables we have used ranges of reflection coefficients obtained from real speech data and a bit allocation based upon the spectral sensitivity properties of the LARs. (These ranges were obtained for 6.7 kHz sampled speech by Lincoln Labs.) Each table lists at the top the minimum and maximum values of the corresponding reflection coefficient, number of bits, and the corresponding LAR step size in dB. We have perturbed the minimum and maximum values supplied by Lincoln Labs a little so that a zero LAR (or equivalently a zero reflection coefficient) is quantized with no error. (Refer to [8] for details.)

The tables are asymmetric (unlike the tables in [2]) insofar as the assumed minimum value of any reflection coefficient is not equal to the negative of its assumed maximum value.

TABLE FOR REFLECTION COEFFICIENT K1

MIN VALUE= -0.960, MAX VALUE= 0.383, NO. OF BITS= 5
LOG AREA RATIO STEP SIZE = 0.636 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-.31446				
	0	-.31348	-13302	-104
-.31243				
	1	-.31130	-13072	-102
-.31008				
	2	-.30878	-12825	-100
-.30739				
	3	-.30590	-12560	-98
-.30430				
	4	-.30260	-12276	-96
-.30077				
	5	-.29881	-11973	-94
-.29672				
	6	-.29449	-11649	-91
-.29210				
	7	-.28955	-11302	-88
-.28683				
	8	-.28394	-10933	-85
-.28085				
	9	-.27756	-10539	-82
-.27406				
	10	-.27034	-10120	-79
-.26639				
	11	-.26220	-9675	-76
-.25775				
	12	-.25304	-9203	-72
-.24805				
	13	-.24278	-8704	-68
-.23722				
	14	-.23136	-8176	-64
-.22518				
	15	-.21868	-7621	-60
-.21186				
	16	-.20471	-7038	-55
-.19722				
	17	-.18939	-6428	-50
-.18123				
	18	-.17273	-5791	-45
-.16389				
	19	-.15473	-5129	-40
-.14524				
	20	-.13544	-4444	-35
-.12534				
	21	-.11495	-3738	-29
-.10429				
	22	-.9338	-3014	-24
-.8224				

(TABLE FOR K1 CONTINUED)

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-8224				
	23	-7089	-2274	-18
-5936				
	24	-4768	-1523	-12
-3587				
	25	-2397	-764	-6
-1200				
	26	0	0	0
1200				
	27	2397	764	6
3587				
	28	4768	1523	12
5936				
	29	7089	2274	18
8224				
	30	9338	3014	24
10429				
	31	11495	3738	29
12534				

TABLE FOR REFLECTION COEFFICIENT K2

MIN VALUE= -0.449, MAX VALUE= 0.956, NO. OF BITS= 5
LOG AREA RATIO STEP SIZE = 0.646 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-14718				
	0	-13729	-4509	-35
-12709				
	1	-11658	-3794	-30
-10580				
	2	-9475	-3060	-24
-8346				
	3	-7196	-2309	-18
-6026				
	4	-4841	-1547	-12
-3642				
	5	-2434	-775	-6
-1219				
	6	0	0	0
1219				
	7	2434	775	6
3642				
	8	4841	1547	12
6026				
	9	7196	2309	18
8346				
	10	9475	3060	24
10580				
	11	11658	3794	30
12709				
	12	13729	4509	35
14718				
	13	15675	5203	41
16598				
	14	17488	5872	46
18342				
	15	19162	6515	51
19947				
	16	20697	7130	56
21412				
	17	22094	7718	60
22742				
	18	23358	8277	65
23942				
	19	24495	8807	69
25018				
	20	25512	9308	73
25979				
	21	26418	9781	76
26832				
	22	27222	10226	80
27588				

(TABLE FOR K2 CONTINUED)

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
27588				
	23	27932	10645	83
28255				
	24	28558	11038	86
28842				
	25	29108	11407	89
29356				
	26	29589	11752	92
29807				
	27	30010	12074	94
30200				
	28	30378	12375	97
30543				
	29	30698	12656	99
30842				
	30	30976	12919	101
31101				
	31	31218	13163	103
31327				

TABLE FOR REFLECTION COEFFICIENT K3

MIN VALUE= -0.911, MAX VALUE= 0.697, NO. OF BITS= 5
LOG AREA RATIO STEP SIZE = 0.650 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-.29856				
	0	-.29641	-11790	-92
-.29410				
	1	-.29164	-11446	-89
-.28900				
	2	-.28618	-11078	-87
-.28318				
	3	-.27997	-10685	-83
-.27655				
	4	-.27291	-10266	-80
-.26904				
	5	-.26492	-9821	-77
-.26054				
	6	-.25589	-9347	-73
-.25097				
	7	-.24575	-8845	-69
-.24023				
	8	-.23441	-8314	-65
-.22826				
	9	-.22178	-7754	-61
-.21497				
	10	-.20781	-7165	-56
-.20030				
	11	-.19245	-6547	-51
-.18424				
	12	-.17568	-5902	-46
-.16677				
	13	-.15751	-5230	-41
-.14791				
	14	-.13799	-4534	-35
-.12774				
	15	-.11720	-3815	-30
-.10636				
	16	-.9526	-3077	-24
-.8392				
	17	-.7236	-2322	-18
-.6060				
	18	-.4868	-1555	-12
-.3663				
	19	-.2448	-780	-6
-.1226				
	20	0	0	0
.1226				
	21	.2448	.780	6
.3663				
	22	.4868	1555	12
.6060				

(TABLE FOR K3 CONTINUED)

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
6060				
	23	7236	2322	18
8392				
	24	9526	3077	24
10636				
	25	11720	3815	30
12774				
	26	13799	4534	35
14791				
	27	15751	5230	41
16677				
	28	17568	5902	46
18424				
	29	19245	6547	51
20030				
	30	20781	7165	56
21497				
	31	22178	7754	61
22826				

TABLE FOR REFLECTION COEFFICIENT K4

MIN VALUE= -0.315, MAX VALUE= 0.822, NO. OF BITS= 4
LOG AREA RATIO STEP SIZE = 0.808 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-10308				
	0	-8915	-2874	-22
-7486				
	1	-6027	-1930	-15
-4543				
	2	-3040	-969	-8
-1523				
	3	0	0	0
1523				
	4	3040	969	8
4543				
	5	6027	1930	15
7486				
	6	8915	2874	22
10308				
	7	11660	3795	30
12969				
	8	14230	4686	37
15442				
	9	16601	5541	43
17707				
	10	18759	6358	50
19756				
	11	20699	7132	56
21589				
	12	22425	7862	61
23210				
	13	23945	8547	67
24631				
	14	25271	9187	72
25867				
	15	26421	9782	76
26934				

TABLE FOR REFLECTION COEFFICIENT K5

MIN VALUE= -0.602, MAX VALUE= 0.547, NO. OF BITS= 4
LOG AREA RATIO STEP SIZE = 0.712 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-19736				
	0	-18859	-6397	-50
-17940				
	1	-16978	-5681	-44
-15975				
	2	-14931	-4935	-39
-13847				
	3	-12725	-4160	-33
-11567				
	4	-10375	-3360	-26
-9151				
	5	-7899	-2539	-20
-6623				
	6	-5324	-1702	-13
-4009				
	7	-2680	-854	-7
-1342				
	8	0	0	0
1342				
	9	2680	854	7
4009				
	10	5324	1702	13
6623				
	11	7899	2539	20
9151				
	12	10375	3360	26
11567				
	13	12725	4160	33
13847				
	14	14931	4935	39
15975				
	15	16978	5681	44
17940				

TABLE FOR REFLECTION COEFFICIENT K6

MIN VALUE= -0,304, MAX VALUE= 0,807, NO. OF BITS= 4
LOG AREA RATIO STEP SIZE = 0,778 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-.9949				
	0	-.8600	-.2770	-22
-.7218				
	1	-.5808	-.1859	-15
-.4376				
	2	-.2927	-.933	-7
-.1467				
	3	0	0	0
.1467				
	4	.2927	.933	7
.4376				
	5	.5808	.1859	15
.7218				
	6	.8600	.2770	22
.9949				
	7	1.1263	.3660	29
1.2537				
	8	1.3768	.4523	35
1.4953				
	9	1.6091	.5354	42
1.7180				
	10	1.8219	.6149	48
1.9208				
	11	2.0146	.6906	54
2.1034				
	12	2.1872	.7623	60
2.2661				
	13	2.3404	.8298	65
2.4100				
	14	2.4752	.8931	70
2.5361				
	15	2.5929	.9522	74
2.6459				

TABLE FOR REFLECTION COEFFICIENT K7

MIN VALUE= -0.551, MAX VALUE= 0.448, NO. OF BITS= 3
LOG AREA RATIO STEP SIZE = 1.198 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-18070				
	0	-16439	-5482	-43
-14691				
	1	-12831	-4197	-33
-10868				
	2	-8814	-2841	-22
-6682				
	3	-4490	-1434	-11
-2256				
	4	0	0	0
2256				
	5	4490	1434	11
6682				
	6	8814	2841	22
10868				
	7	12831	4197	33
14691				

TABLE FOR REFLECTION COEFFICIENT K8

MIN VALUE= -0.286, MAX VALUE= 0.570, NO. OF BITS= 3
LOG AREA RATIO STEP SIZE = 1.023 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-.9380				
	0	-.7580	-.2435	-19
-.5730				
	1	-.3842	-.1226	-10
-.1928				
	2	0	0	0
.1928				
	3	.3842	.1226	10
.5730				
	4	.7580	.2435	19
.9380				
	5	1.1121	.3612	28
1.2793				
	6	1.4390	.4742	37
1.5907				
	7	1.7339	.5816	45
1.8685				

TABLE FOR REFLECTION COEFFICIENT K9

MIN VALUE= -0.406, MAX VALUE= 0.504, NO. OF BITS= 3
LOG AREA RATIO STEP SIZE = 1.069 DB

X(J)	J	R(J)	INDEX(J)	INDEXP(J) (X 2**7)
-13306				
	0	-11581	-3768	-29
-9779				
	1	-7909	-2543	-20
-5983				
	2	-4014	-1281	-10
-2015				
	3	0	0	0
2015				
	4	4014	1281	10
5983				
	5	7909	2543	20
9779				
	6	11581	3768	29
13306				
	7	14948	4941	39
16499				

V. INTERPOLATION AND SYNTHESIS

We suggest that at the receiver the decoded parameters be (linearly) interpolated between parameter receptions. When the transmission interval for a parameter exceeds 100 msec, it is recommended that the last received parameter value be repeated. This issue was discussed in more detail in Section II-B.

For the implementations using the SPS-41/PDP-11 system, programs may be written for the PDP-11 to supply to the SPS-41 interpolated parameters at intervals of IFI msec, which the SPS-41 can further interpolate to update the synthesizer parameters at smaller intervals (e.g. every 4.8 msec).

VI. SUMMARY OF SPECIFICATIONS

Analysis is done every IFI=9.6 msec. An LPC order of M=9 is recommended. VFR transmission of reflection coefficients is accomplished using the basic likelihood ratio method, where the threshold LRT=1.4. Pitch and gain are transmitted at a fixed rate of every 19.2 msec. During an unvoiced region, only the first pitch value (=0) is transmitted. New coding/decoding tables are employed to quantize pitch, gain and reflection coefficients with 6, 5 and 36 bits respectively. A parcel of data bits with a 3-bit header is transmitted every 9.6 msec. A

variable-length packet representing a maximum speech duration of 400 msec is recommended. Parameter interpolation between transmissions is suggested.

For the specified VFR transmission scheme, the average frame rate for reflection coefficients is about 37 frames/sec; that for gain is 52 frames/sec; that for pitch is less than about 40 frames/sec. A reasonable estimate of the average frame rate for all the transmission parameters is about 40 frames/sec. This corresponds to a data rate of 40(5+6+36)=1880 bps. The bit rate due to the 3-bit parcel overhead is 104x3=312 bps. Thus, we estimate the average bit rate to be on the order of 2200 bps for continuous speech. Explicit silence detection as being done in System I is expected to drop this rate to about 1000 bps or less depending upon the proportion of silence relative to speech.

VII. OTHER GENERAL RECOMMENDATIONS

A. Gain Implementation

We recommend implementing the speech signal energy as a gain multiplier at the input of the synthesizer filter. With the gain multiplier placed at the output of the filter, perceivable distortions are produced in the synthesized speech at places where relatively large frame-to-frame

energy changes occur [1]. (There are, however, adhoc solutions to this problem).

B. Future System Updates

As mentioned in the introduction, our objective in coming up with specifications for System II has been to procure maximum benefit with minimum effort. In keeping with this objective, we left out the bit-saving techniques: variable order linear prediction, Huffman or other (suboptimal) fancy encoding (e.g. delta coding of pitch or gain) [4] and the optimal linear interpolation scheme which holds potential for improving speech quality especially with VFR transmission [9]. We suggest that these techniques, and perhaps others as well, be considered for a future System III.

REFERENCES

1. BBN Quarterly Progress Report on Command and Control Related Computer Technology, Report No. 3093, Part II, June 1975.

2. D. Cohen, "Specifications for the Network Voice Protocol (NVP)," NSC Note 68, Nov. 1975.

3. D. T. Magill, "Adaptive Speech Compression for Packet Communication Systems," Proc. Nat'l Telecommun. Conf., pp. 29D-1 - 29D-5, Nov. 1973.

4. J. Makhoul, R. Viswanathan, L. Cosell and W. Russell, Natural Communication with Computers, Final Report, Vol. II, Speech Compression Research at BBN, Report No. 2976, Dec. 1974.

5. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. ASSP, Vol. SP-23, pp. 67-72, Feb. 1975.

6. J. Makhoul and L. Cosell, "Recommendations for Encoding and Synthesis," NSC Note 49, Nov. 1974.

7. J. Markel, "Proposal for NSC-LPC Coding/Decoding Tables," NSC Note 36, July 1974.

8. R. Viswanathan and W. Russell, "Quantization Routines for Linear Predictive Vocoders," NSC Note 33, July 1974.

9. R. Viswanathan, J. Makhoul and W. Russell, "Optimal Linear Interpolation in Linear Predictive Vocoders," NSC Note 59, April 1975.

APPENDIX D

EFFECT OF LOST PACKETS ON SPEECH INTELLIGIBILITY

NSC Note 78, February 24, 1976

(Author: A.W.F. Huggins)

1. INTRODUCTION

So far, the decision on how much speech a packet should contain for transmission over the ARPA net has been influenced by two main factors: overhead, and delay. In the present implementation, each packet contains a maximum of 1007 data bits, of which about 32 are needed for overhead. An additional 200 bits of overhead (not included in the 1007) are added by the IMP. The speech data consists of 67 bit parcels, each of which encodes 19.2 msec of speech. (These values may change in future systems). The more parcels a packet contains, the smaller the percentage of bits "wasted" in overhead. This factor argues for maximizing the number of parcels in each packet. On the other hand, increasing the number of parcels per packet increases the duration of speech encoded in the packet. Since the first parcel in the packet cannot be transmitted until the last parcel in the same packet has been encoded, a delay is unavoidably introduced, equal to the duration of speech encoded in a packet. This delay is in addition to delays due to other factors such as finite transmission time, path length, and network response. Delays have a serious disruptive effect on conversation (Riesz and Klemmer, 1966; Brady, 1971), and this argues for minimizing the duration of speech in a packet. Experiments have been performed with two choices of speech duration per package. ISI has used the maximum number of parcels per packet (14) corresponding to 268.8 msec of speech, yielding an overhead

rate of 17.5%. Lincoln Labs, on the other hand, has used up to 7 parcels per packet, corresponding to 134.4 msec of speech, and an overhead of 29.8%.

The purpose of this note is to argue that a third factor needs to be considered in deciding how much speech should be encoded in one packet - the effect of lost packets on intelligibility. We propose a method of packetizing speech parcels which will sharply reduce the effect on speech intelligibility of lost (delayed) packets.

2. THE PROBLEM

Whenever an utterance is longer than the typical processing and transmission delays, reconsti ution of the waveform begins at the destination before the message ends at the transmitter. Since packets must be reconstituted in the correct sequence, and the sequence has already begun, a problem arises whenever a packet is delayed. Two solutions have been tried. Lincoln Labs has chosen to proceed without the late packet, replacing the speech in the late packet by an equal amount of silence. This solution discards some of the speech waveform, but retains the overall temporal pattern of the speech. ISI has chosen to wait for the late packet, thus introducing a silence equal to the delay between the expected and actual arrival times of the delayed packet (a variable). This solution does not discard any of the speech waveform, but

the overall temporal pattern of the utterance may be disturbed. As network traffic becomes heavier, the interruptions introduced into the speech by the former solution, and the long delays introduced by the latter, become increasingly objectionable.

At the ARPA Review meeting in Reston, Virginia, December 15-16, 1975, Jim Forgie played some packet-speech that had been sent over the ARPANET, for a variety of packet loss rates ranging from 30% to values close to zero. Speech intelligibility was severely affected by 30% loss rates, and substantially affected by loss rates of a few percent. Earlier work on the degradation of intelligibility as a result of interrupting speech (Huggins, 1964), or introducing silent intervals into it (Huggins, 1975a), has shown that the degradation is critically dependent on the duration of the resulting silent intervals. The most severe degradation occurred when the silent intervals lasted 100-300 msec, but intelligibility was much less affected by shorter silent intervals. Thus it appears that the present choice of speech duration per packet leads to silent intervals (due to lost packets) that fall in the range that maximally degrade intelligibility. We summarize the earlier work below, before proposing a remedy, and tests to validate it.

2.1 Interrupted Speech.

The stimulus materials in both the earlier studies were continuous speech, consisting of readings from a book of scientific essays. Intelligibility was measured by the number of words in 100-word passages that listeners were able to repeat correctly in a shadowing task, where the listener repeats aloud, word for word, what he hears. Subjects were run individually. The stimulus tapes for the interrupted speech experiments were generated by switching the continuous speech message backwards and forwards between two tape recorders at a regular rate, so that the signal deleted by an interruption on one tape always appeared on the other tape. The two interrupted tapes thus produced were therefore complementary. Switching rates varied between one-fifth and sixteen complete cycles of alternation per second, and the speech-silence ratio was equal to 1.0 on each tape. Thus, silent intervals (and speech intervals) ranged in duration from 2500 msec down to 31 msec on each tape. Twenty subjects each shadowed one of the two tapes. At the slowest switching rate, subjects heard half the phrases, and intelligibility was about 50%. As the rate was increased, intelligibility first declined to a minimum of 15-20%, with speech and silent intervals between 300 and 100 msec, and then improved rapidly to 80% with silent intervals of 31 msec. (See Fig. 1). Thus, intelligibility was most degraded when speech and silent intervals lasted 100-300 msec, but was little affected when

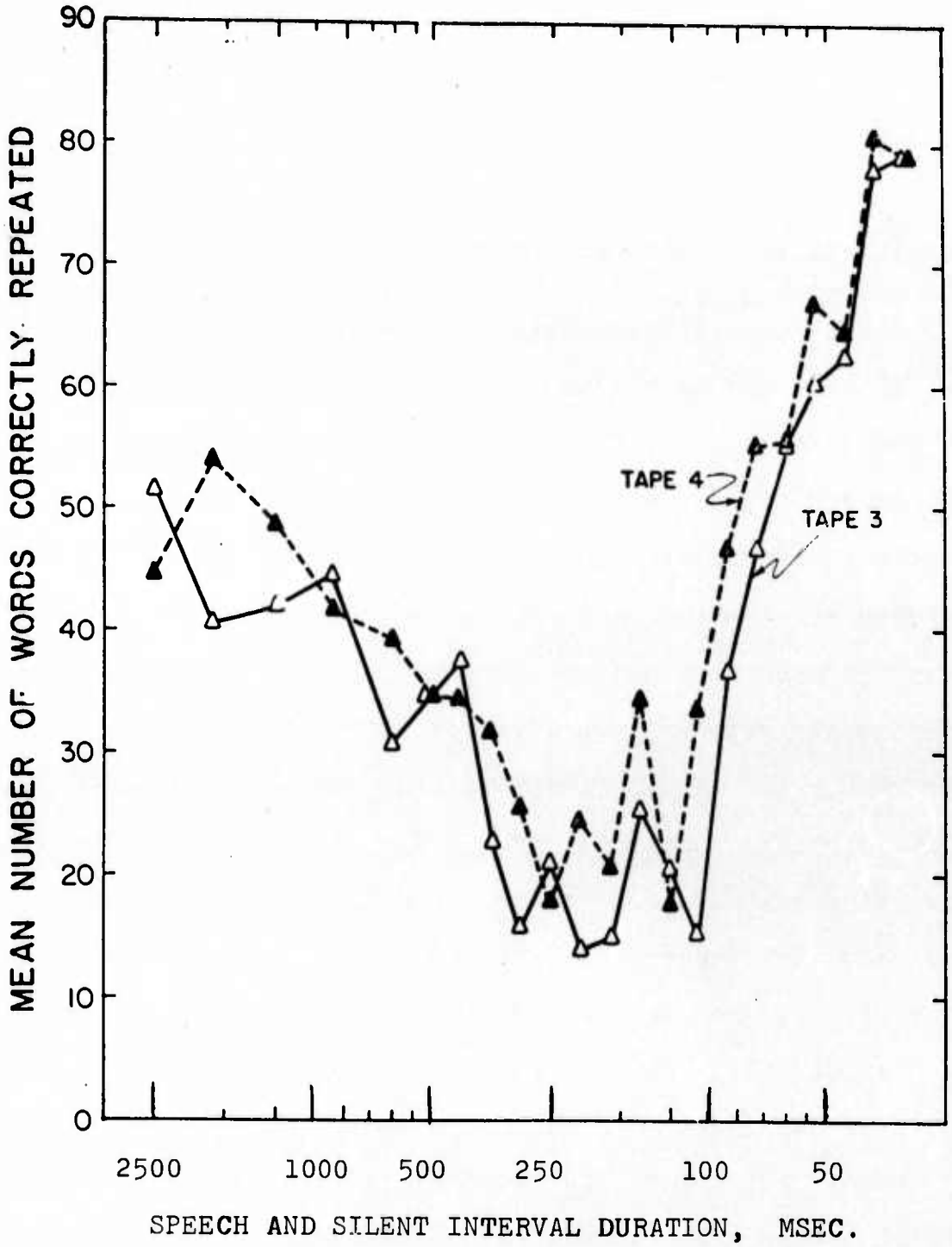


Figure 1. Shadowing scores as a function of speech and silent interval duration for two complementary interrupted speech tapes. (From Huggins, 1964.)

speech and silent intervals were shortened to 31 msec, even though 50% of the speech was missing.

2.2 Temporally Segmented Speech.

The temporally segmented speech experiments differed from the interrupted speech experiments only in that no speech was discarded (Huggins, 1975a). Instead, the continuous speech message was broken up into "speech intervals" by the insertion of silent intervals. Similar effects could be obtained by repeatedly starting and stopping a tape recorder, if the transport mechanism had no inertia. The durations of speech and silent intervals were varied independently. The results show that, with silent intervals held constant at 200 msec, intelligibility declined from 95% to less than 20% as speech interval duration was decreased from 200 msec to 30 msec. (See Fig. 2, Curve A). On the other hand, with speech intervals held constant at 63 msec, intelligibility remained low (about 50%, the level depending only on speech interval duration) as silent intervals were shortened from 500 msec to 125 msec, then suddenly and rapidly recovered as silent intervals were reduced from 125 to 63 msec. At 63 msec or below, intelligibility was close to 100% (See Fig. 2, Curve B).

These results strongly support the hypothesis that the V-shaped minimum of intelligibility found in a variety of

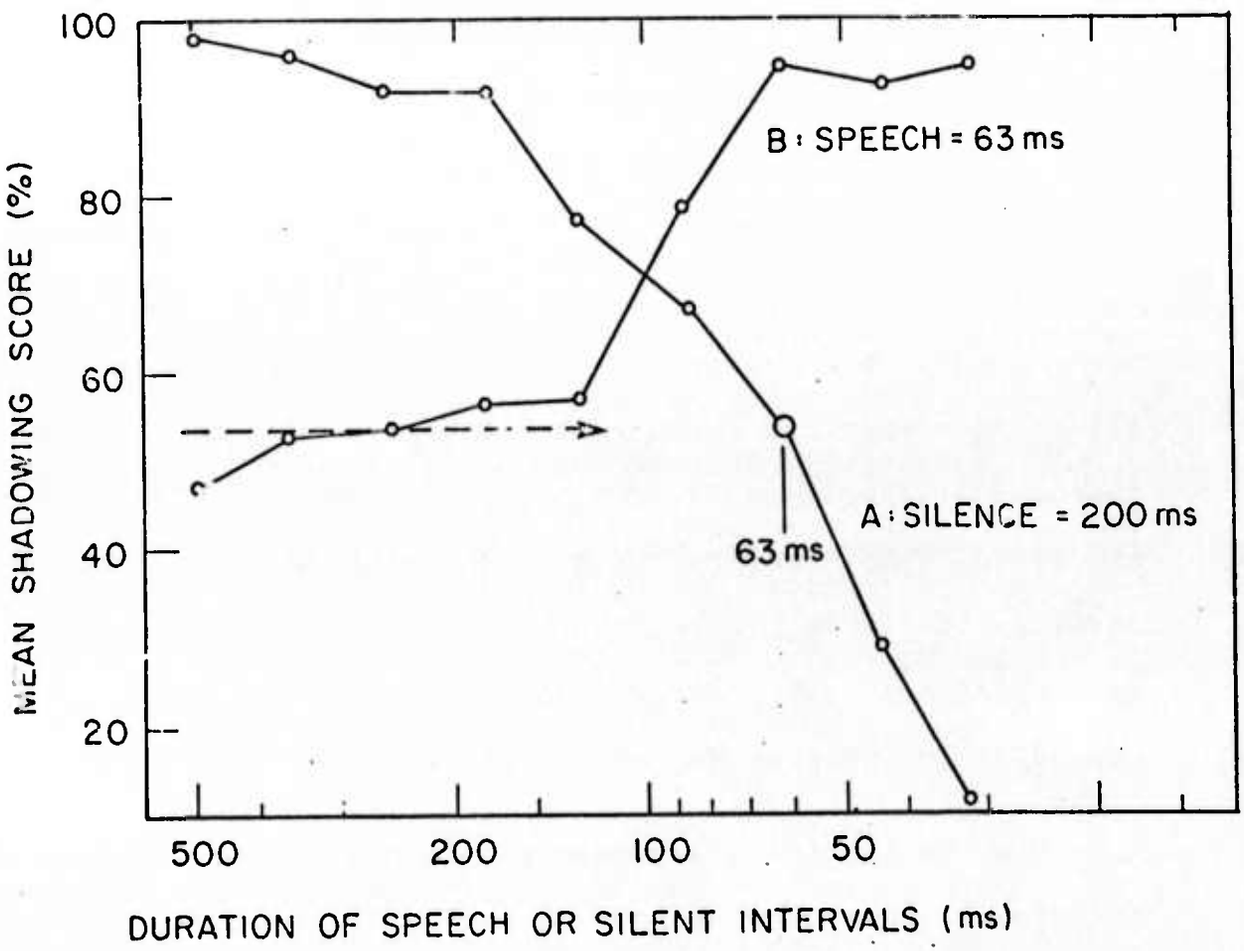


Figure 2. The intelligibility of temporally segmented speech (shadowing scores) as a function of speech interval duration (Curve A: silence fixed at 200 msec), and as a function of silent interval duration (Curve B: speech fixed at 63 msec). (From Huggins, 1975.)

experiments of this sort, of which Figure 1 is an example, is produced by the overlap of two separate effects. The decline of intelligibility as speech and silent interval durations are shortened towards 100 msec is due to the decreasing amount of information in the speech intervals, together with the fact that the silent intervals are too long for the ear to be able to "bridge" them. Other experiments (Huggins, 1974; Wingfield and Wheale, 1975) have shown that this decline is affected by speech rate, and the variable defining the decline is the amount of speech in each speech interval (i.e. the number of syllables, phonemes, etc) rather than its duration. On the other hand, the recovery of intelligibility as speech and silent intervals are further shortened is due to the ear's increasing ability to bridge the silent intervals as they are shortened. The recovery due to the gap-bridging takes place despite the progressive decline of intelligibility of the speech intervals, as they are shortened. The recovery is not dependent in the same way on speech rate (Huggins, 1975b).

How are the foregoing experiments related to the effects of lost speech packets? At present, each lost packet introduces a silent interval lasting 135-270 msec. These silences are too long for the ear to bridge. As long as their rate of occurrence is low they have only a small effect on intelligibility, since the intervals of speech occurring between successive silences tend to be quite long. As the rate of lost packets increases, the duration of intact speech

intervals declines, with serious effects on intelligibility.

The tasks in the foregoing experiments are quite similar to conditions a vocoder user might actually encounter. The shadowing task can be thought of as increasing the processing load on the listener. Although a real-life user would not normally repeat all he heard, word-for-word, and might therefore better understand the more difficult passages, he might easily have other secondary tasks to perform, or be operating under adverse conditions, which could produce increases in processing load similar to those induced by the shadowing task.

There are, however, two aspects of the tasks that are not very realistic. First, the silent intervals were regularly spaced in time, whereas one would expect late-arriving packets to occur randomly in time. However, two earlier studies suggest that randomly timed deletions would produce intelligibility decrements similar to those obtained with regular deletions. Miller and Licklider (1950) reached this conclusion in their study of the intelligibility of PB word lists subjected to regular and to random interruptions, and Cherry (1953) mentions the same conclusion in his first study of speech alternated between the ears. (See Huggins (1964) for arguments that alternated and interrupted speech show reduced intelligibility for the same reason).

Secondly, the proportion of speech discarded in the interrupted speech experiment described above was 50%, and it is unlikely that packet loss rates on the ARPANET would ever be this high. On the other hand, Jim Forgie's demonstration at the Reston meeting showed that intelligibility can be affected by even quite low loss rates.

3. A REMEDY

The most obvious remedy for the problem of lost packets is to increase the redundancy of transmission, so that speech parcels do not get lost. Two obvious ways of increasing redundancy are, 1) to transmit each packet twice, and 2) to arrange that each parcel of speech is transmitted in two different packets. These procedures effectively square the probability of a lost packet, but at a cost of raising the overhead to a minimum of 58.7%, since one of every two packets contains no new information.

There are other possibilities. All the studies mentioned above agreed in the conclusion that the disruption of intelligibility becomes less severe as the duration of the silent intervals is reduced. The ideal way of reducing the intelligibility deficit, resulting from lost packets, is to substitute the loss of parcels for the loss of packets. The loss of a single parcel results in a silence of 19.2 msec, which produces a negligible effect on intelligibility, even at

high loss rates.

There are two ways to achieve the replacement of lost packets by lost parcels. One is simply to equate parcels and packets, transmitting a single parcel in each packet. This would virtually eliminate the intelligibility loss, even at loss rates approaching 50%. Note also that this solution would almost eliminate that part of the speech-input-to-speech-output delay generated during coding and packing the speech for transmission. The cost, again, is in greatly reduced efficiency of transmission. About 75% of transmitted bits would be overhead, if every packet contained only a single parcel. This remedy is therefore less efficient than transmitting each packet twice.

A way of reducing the overhead costs of both the foregoing solutions (repeating packets, and one parcel per packet) would be to adopt the less efficient procedure only when packet loss rates are becoming objectionably high, perhaps under feedback control of the receiver. A disadvantage of this approach is that the most probable reason for a packet being delayed is that the net is being heavily used (a situation increasingly likely as time progresses). Yet the suggested solution aggravates the situation by <u>increasing</u> the net traffic, since it uses a less-efficient transmission scheme.

4. PROPOSED SOLUTION.

A second way of replacing lost packets by lost parcels is to distribute the parcels between several packets in such a way that loss of a packet does not result in loss of adjacent parcels. This could be achieved by interleaving - that is, by transmitting odd-numbered parcels in one packet, and even-numbered parcels in a second. The loss of one packet would then result in a brief burst of interrupted speech, at a rate of 25 interruptions per second, which would (extrapolating from Figure 1) have a negligible effect on intelligibility, even at quite high loss rates.

The proposed solution does not increase the overhead, since it effectively takes advantage of the redundancy inherent in the speech waveform, rather than adding redundancy deliberately. It effectively squares the probability that a lost packet will result in a silent interval, since the loss of one packet results in a burst of interrupted speech, and two sequential packets must be lost for a silent interval to occur.

There is one condition under which none of the foregoing redundancy adding schemes would work. If the probability of a packet being delayed was not independent of the fate of other packets, the chance of two adjacent packets being delayed might be close to the chance of a single packet being delayed. This could easily happen if the reason for a packet being

delayed was that the traffic load on the net had briefly reached its full capacity. Then all subsequent packets would be held up until the net overload eased. The number of packets held up would depend on the duration of the overload. The interleaving scheme does provide a possible solution even to this problem, up to a loss of perhaps three adjacent packets: increase the depth of interleaving, by distributing parcels between (say) four separate packets instead of two. This solution quickly runs into diminishing returns, since intelligibility begins to fall when silent intervals are longer than about 60 msec. The loss of three adjacent packets, interleaved to depth four, would result in one parcel of speech followed by three parcels of silence, repeated cyclically for the duration of a packet. It may be, however, that the situation that requires interleaving to depth greater than two may not arise. Measurements of packet delays have shown (Forgie, personal communication) that the probabilities of adjacent packets being delayed are independent, at least with present network loads.

A disadvantage of interleaving is that, for a given number of parcels per packet, the duration of speech coded in the packet is increased by a factor equal to the depth of interleaving. However, this would probably not introduce unacceptable difficulties, as long as the depth of interleaving did not exceed two. It could be counteracted by reducing the number of parcels per packet, at the cost of

increased overhead.

In the interleaving scheme outlined above, odd-numbered parcels are transmitted in one packet, and even-numbered parcels in a second. This is diagrammed in Figure 3a, where each digit represents a parcel. The first six odd-numbered parcels are transmitted in the first packet, and the first six even-numbered parcels in the second. There is a temporal offset of one parcel between packets 1 and 2, but an offset of 11 parcels between packets 2 and 3. There are some advantages to staggering the interleaved packets, so that the first parcel of the later packet slots into the middle, rather than the start, of the preceding packet. The staggered interleaving scheme is diagrammed in Figure 3b. In the former scheme, packets become ready for transmission in pairs, which maximizes the chance of both packets being delayed if network overload is the cause of delay. Thus, packet 2 is ready for transmission one parcel after packet 1, but packet 3 is not ready until 11 parcels after packet 2 (with six parcels per packet). In the staggered scheme, this risk is reduced, since each packet becomes ready for transmission either five or seven parcels after the preceding packet.

A second advantage of a staggered scheme of interleaving is that the decision to proceed without a packet can be reviewed at the start of the next new packet. If the late packet has arrived by then, the later parcels in the late

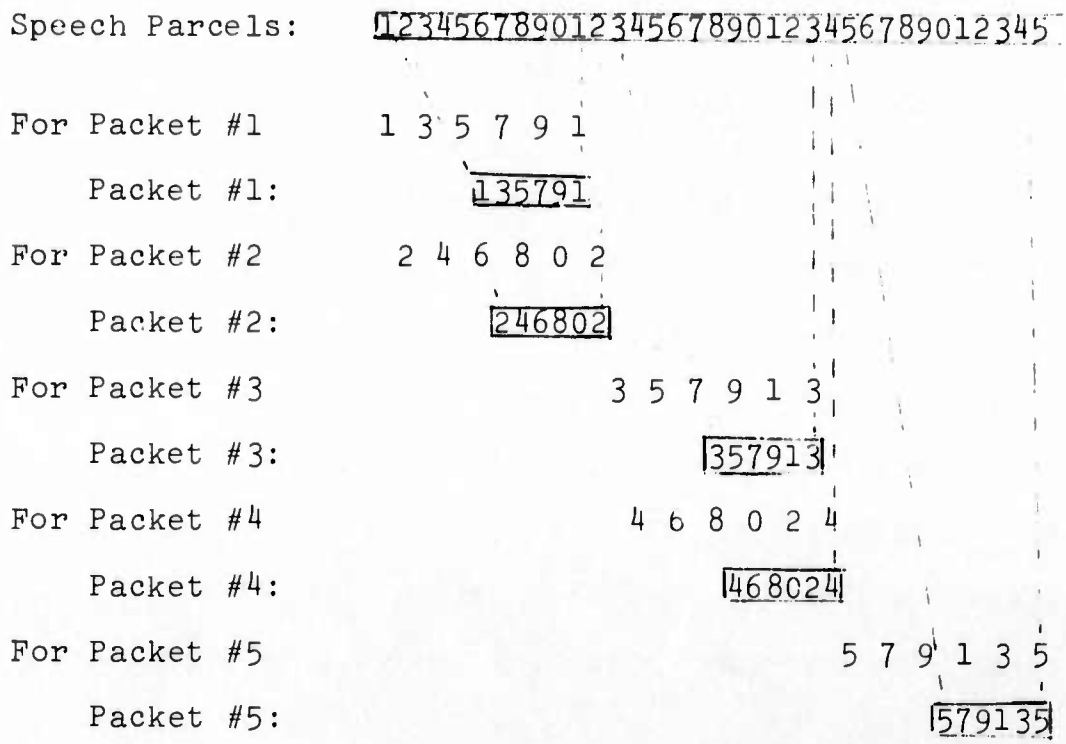


Figure 3a: Simple Interleaving.

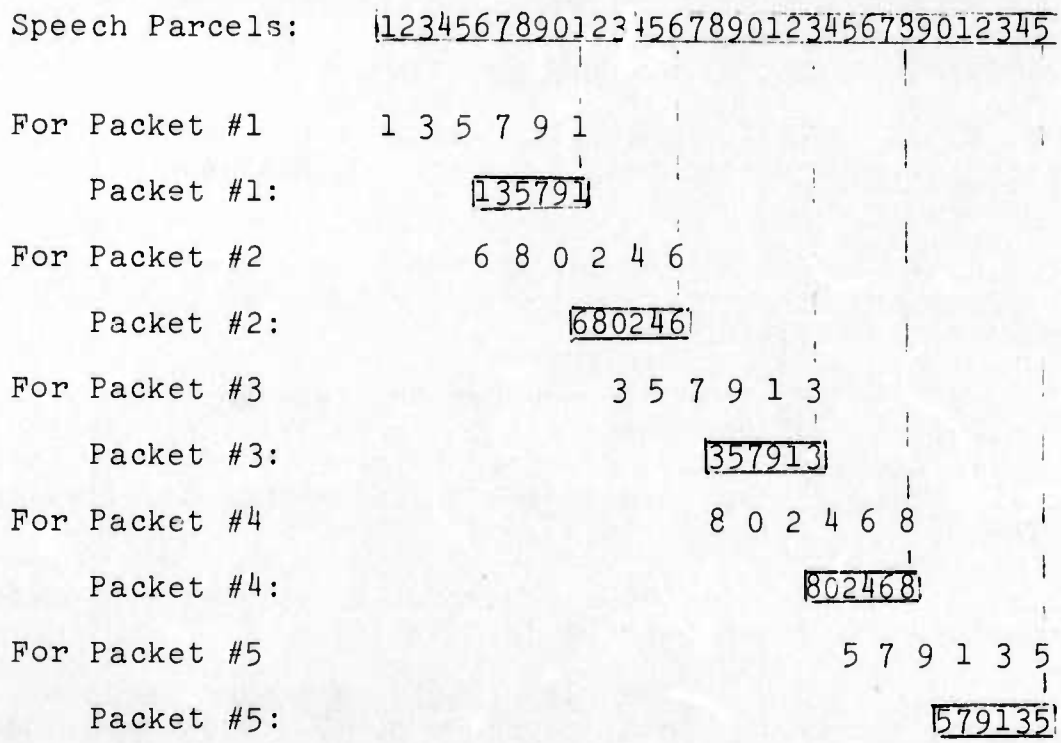


Figure 3b: Staggered Interleaving.

packet can be incorporated in the reconstituted speech. This procedure would often halve the duration of interrupted speech introduced by a late packet.

We propose to run intelligibility tests, using the IEEE recommended sentences, to test the correctness of the foregoing arguments. The simplest method of performing the tests is to acquire recordings of the sentences that have already been passed through a variety of vocoding systems, and then simulate the effects of lost packets, and lost interleaved packets, by appropriate analog switching of the waveform. Any comments or suggestions will be appreciated.

5. REFERENCES.

Brady, P. T., (1971) Effects of transmission delay on conversational behavior on echo-free telephone circuits. Bell Syst. Technical Journal 50, 115-134.

Cherry, E. C., (1953) Some experiments on the recognition of speech, with one and with two ears. J.Acoust.Soc.Amer.25, 975-983.

Huggins, A. W. F., (1964) Distortion of the temporal pattern of speech: interruption and alternation. J.Acoust. Soc.Amer.36, 1055-1064.

Huggins, A. W. F., (1974) More temporally segmented speech: is duration or speech content the critical variable in its loss of intelligibility? Research Laboratory of Electronics, Quart. Prog. Rep. 114, 185-193, Massachusetts Institute of Technology, July 15, 1974.

Huggins, A. W. F., (1975a) Temporally segmented speech. Perception and Psychophysics 18, 149-157.

Huggins, A. W. F., (1975b) Temporally segmented speech and "echoic" storage. In A. Cohen & S. G. Nooteboom, Eds., Structure and Process in Speech Perception. Springer-Verlag, New York 1975.

Miller, G. A., & Licklider, J. C. R., (1950) The intelligibility of interrupted speech. J.Acoust.Soc. Amer.27, 167-173.

Riesz, R. R., & Klemmer, E. T., (1966) Subjective evaluation of delay and echo suppressors in telephone communications. Bell Syst. Technical Journal 45, 2919-2941.

Wingfield, A., & Wheale, J. L., (1975) Word rate and intelligibility of alternated speech. Perception and Psychophysics 18, 317-320.

APPENDIX E

INSTRUCTIONS TO HIGH SCHOOL SUBJECTS

1) We are doing research on ways to transform speech into numbers so that people can speak to computers, and so that computers can repeat the message to others, while sounding just like the original speaker.

2) The approach requires transforming speech sounds into strings of numbers.

3) That is not difficult. For example, take an electrical signal from a microphone, measure the voltage and feed the voltage readings into the computer.

4) The problem is that in order to end up with computer speech that is sharp and clear, and sounds like the original human speaker, a very fine record of the voltage changes is required. It takes thousands of numbers to represent just one little word.

5) What we are trying to do is find ways of taking away a lot of the numbers without affecting the clarity or recognizability of the words.

6) Today we want to see how successful some of these approaches are.

7) We will have you listen to some words spoken* by a computer. *Actually the computer puts out voltage readings which drives a Hi Fi set. Sometimes the words will be sharp and clear, and sometimes they will be very difficult to hear.

8) Because you might be able to recognize familiar words even if they are unclear, we will use artificial words.

9) They will be very short words like:

T	U	P
G	U	K
Z	I	M
S	I	Z

10) We will tell you the vowel in the middle. You will select the consonants on one or both sides.

11) Lets do some examples:

A) For this list there is a single set of possible consonants

. The consonants are b d g v z zh

. The sound of each is familiar except perhaps for zh - as in azure.

. The vowels are ah as in (father)
ih as in (bit)

. The first item will have ih's in the middle

. When I say the word, listen for the first and last consonant.

. Tell me the first consonant by circling it in the left string on the answer sheet.

. Tell me the final consonant by circling it in the right string on the answer sheet.

. Every word will be preceded by ah

. Read*

B) Slightly different situation

. String of possible first consonants different from final consonants

. Sounds of consonants familiar except perhaps y as in (yet) and ng as in (sing)

. Vowels ah as in (father), ih as in (sing)

. This time we will do 6 items in a row

. Write down clock-count you see on clock after you have circled final consonant for each item. Put clock-count in space to right of each item.

C) Still different situation

. There is just a first consonant

. Vowels i as in (beat), ah as in (father)

. Lets do six items, 5 seconds apart

. Write down time after circling the consonant

*Check Answer Sheet (C)

12) . You will have other lists as well as these

.Just check the heading for consonant sounds, vowel sounds.

.All items will be 5 seconds apart

13) Be as accurate as possible, but be as fast as possible.

14) Take as much time as you need to be as sure as you ever will be, but take absolutely no more time than you have to.

15) We are very interested in whether it takes longer to hear some of these words than others.

16) To show differences in hearing time, you have to respond as quickly as possible.

16a) What number to mark. Number you are sure must have been on clock when you looked up.

16b) Write time first, then fix mistakes.

17) Now having said that: I don't want you to blow a gasket trying to be super good - at the start - and then be so wrung out that you do a bad job at the end. This will be a long session, it may get to be pure drudgery. Please try to adopt a level of tension/effort that will carry you through to the bitter end operating at an effective level.

18) Just because some items sound like you heard them before, don't assume they are same or if same, that your prior response was right, i.e. make independent judgements on each item.

19) We will take a break about half way through, cokes on the house.